SQLintersection

Session: 11/21, 10am – 11:15am

SQL Server's Path Toward an Intelligent Database

Pedro Lopes
@SQLPedro

SQL intersection

Microsoft

Speaker: Pedro Lopes

- Program Manager @ Azure Data SQL Server team
- Relational Engine: Query processing; Performance
- Compatibility Certification (https://aka.ms/dbcompat)


/pedroazevedolopes


@SQLPedro

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 five-day SQLskills Immersion Event!
  - The first round of drinks is sponsored by SentryOne and SQLskills

SQL intersection

SQLskills immerse yourself in sql server & SentryOne™

Overview

- **The Intelligent Database vision**
- **Intelligent Query Processing scenarios**
  - Demos

The Intelligent Database vision

Intelligent Database

Anxiety-free upgrades

- Database compatibility level regression protection
- Automatic plan regression correction
- Query Tuning Assistant

Intelligent performance

- Intelligent Query Processing
- Resource governor
- In-Memory Databases
- Auto-Soft NUMA

Management-by-default

- Automatic Indexing in SQL DB
- Flexible Scaling
- Integrity Checking
- Intelligent Insights
- Lightweight Query Profiling
- Accelerated Database Recovery

Security in-depth 24x7

- Advanced Threat Detection
- Data Classification
- Vulnerability Assessment

Adapts to the constantly changing world of businesses and data

Intelligent Database

Anxiety-free upgrades

- Database compatibility level regression protection
- Automatic plan regression correction
- Query Tuning Assistant

Intelligent performance

- Intelligent Query Processing
- Resource governor
- In-Memory Databases
- Auto-Soft NUMA

Management-by-default

- Automatic Indexing in SQL DB
- Flexible Scaling
- Integrity Checking
- Intelligent Insights
- Lightweight Query Profiling
- Accelerated Database Recovery

Security in-depth 24x7

- Advanced Threat Detection
- Data Classification
- Vulnerability Assessment

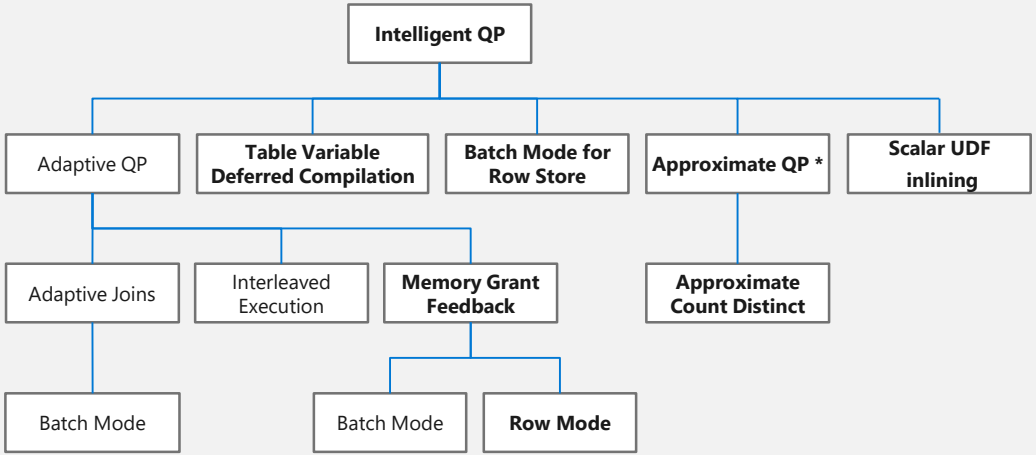
Adapts to the constantly changing world of businesses and data

Intelligent Query Processing with SQL Server

The Intelligent Database

- Available by default on the latest database compatibility level setting
- Delivering broad impact that improves the performance of existing workloads with minimal implementation effort
- Critical parallel workloads improve when running at scale, while remaining adaptive

The Intelligent Query Processing feature family



Bold indicates new and improved features in SQL Server 2019

Itaú-Unibanco

"As early adopters of SQL Server 2019, we are really pleased with the performance and results we achieved from using the new SQL Server features. **The Intelligent Query Processing feature has increased the speed and processing power of our business. We are now able to achieve significant performance improvements, faster applications with minimum downtime all without any changes to the lines of code.** With Accelerated Database Recovery, long-running application transactions can rollback and be recovered in a matter of seconds which is a huge relief for our database administrators, saving us significant time and resources. We feel safe and can confidently look forward to providing even more satisfaction to our customers."

Edilson Albuquerque, Database Team Manager, Itaú-Unibanco

140 and 150 database compatibility level

Memory Grant Feedback (MGF)

Queries may spill to disk or take too much memory based on poor cardinality estimates. Memory misestimations result in spills, and overestimations hurt concurrency

MGF will adjust memory grants based on execution feedback

Batch Mode in 140, Row Mode in 150

MGF will remove spills and improve concurrency for repeating queries

- Spills to disk → MGF corrects grant misestimations
- Excessive memory grant → MGF corrects wasted memory, improves concurrency

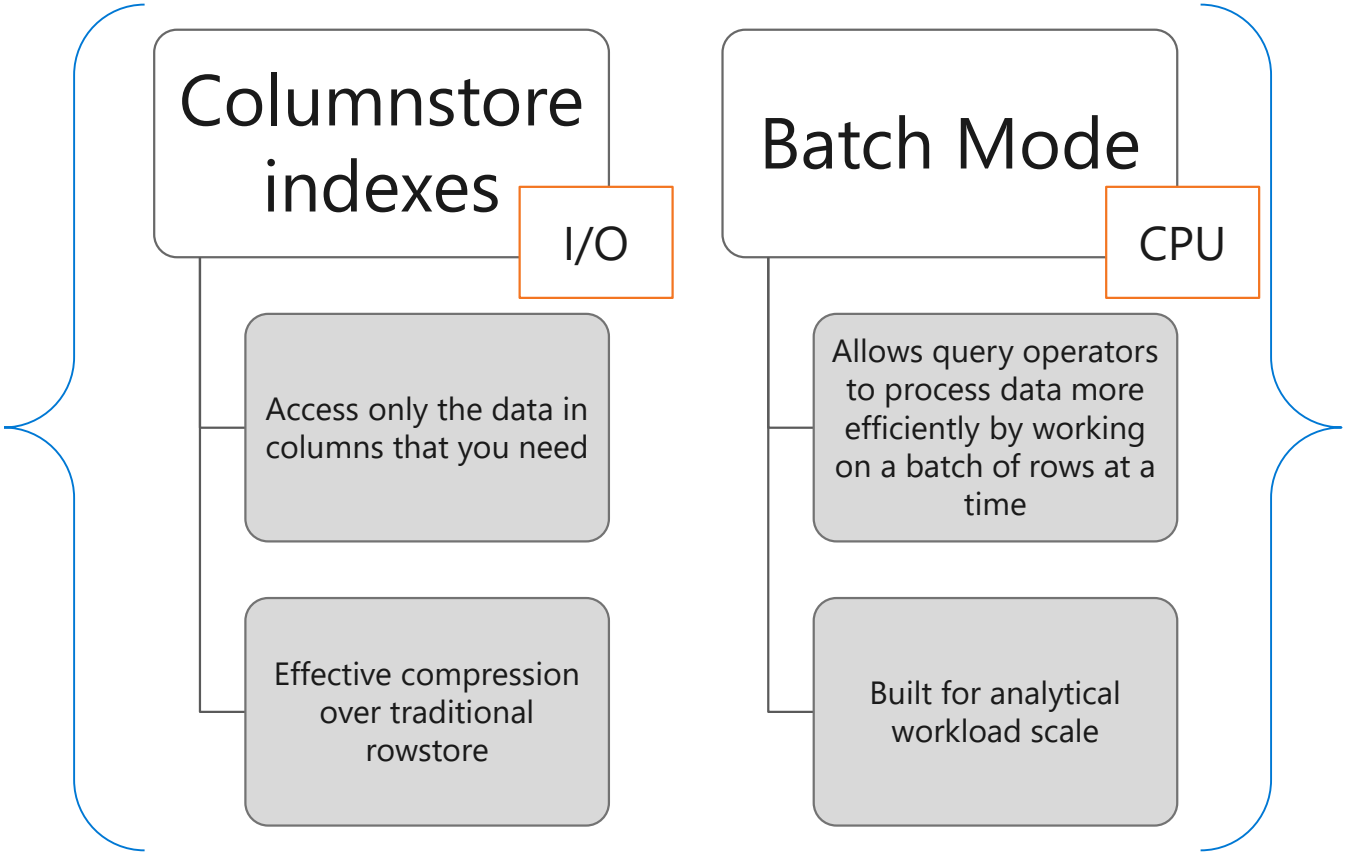
Demo

Memory Grant Feedback

SQL intersection

Batch Mode and Columnstore

Since SQL Server 2012 we've bound these two features together



150 database compatibility level

Batch Mode on Rowstore

Sometimes Columnstore isn't an option:

- OLTP-sensitive workloads
- Vendor support
- Columnstore interoperability limitations

Now get analytical processing CPU-benefits **without Columnstore indexes.**

Batch mode on rowstore supports:

- On-disk heaps and B-tree indexes and existing batch-capable operators (**new scan operator** can evaluate batch mode bitmap filters)
- Existing batch mode operators

Batch Mode on Rowstore candidate workloads

A significant part of the workload consists of analytical queries **AND**

The workload is CPU bound **AND**

- Creating a columnstore index adds too much overhead to the transactional part of your workload **OR**
- Creating a columnstore index is not feasible because your application depends on a feature that is not yet supported with columnstore indexes **OR**
- You depend on a feature not supported with columnstore (for example, triggers)

Demo

Batch Mode on Rowstore

SQL intersection

GVC Holdings PLC

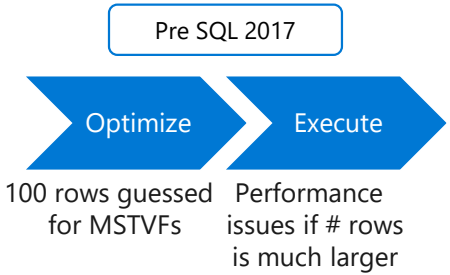
"We will continue using In-Memory OLTP to provide wicked-fast solutions for pure caching and persisted hot and cold caching. We are now looking forward to embracing the new Intelligent Query Processing feature in SQL Server 2019 for our typical OLTP workloads, especially batch mode on row store and memory grant feedback."

Milos Radivojevic - Senior Database Developer MSSQL, GVC Holdings PLC

Up to 130 database compatibility level

Interleaved Execution for MSTVFs

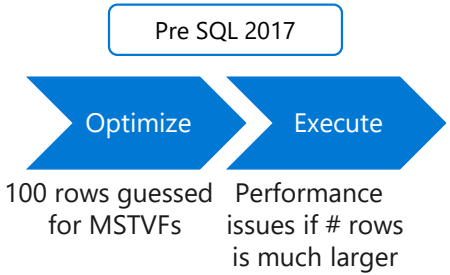
Multi-statement table-valued functions (MSTVFs) are treated as a black box by QP and SQL Server uses a fixed optimization guess.



140 database compatibility level

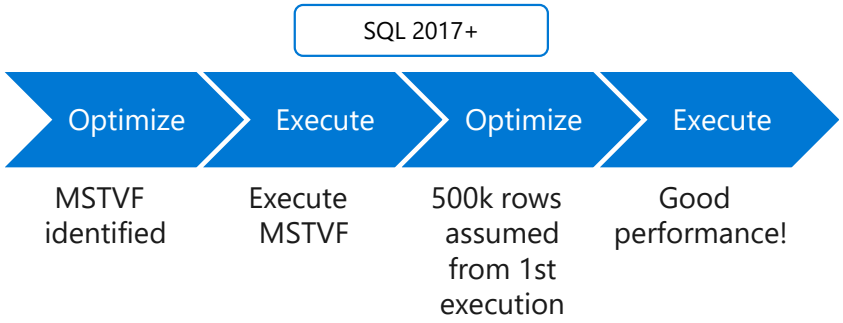
Interleaved Execution for MSTVFs

Multi-statement table-valued functions (MSTVFs) are treated as a black box by QP and SQL Server uses a fixed optimization guess.



Interleaved Execution will materialize and use row counts for MSTVFs.

Downstream operations will benefit from the corrected MSTVF cardinality estimate.



Demo

Interleaved Execution for MSTVFs

SQL intersection

140 database compatibility level

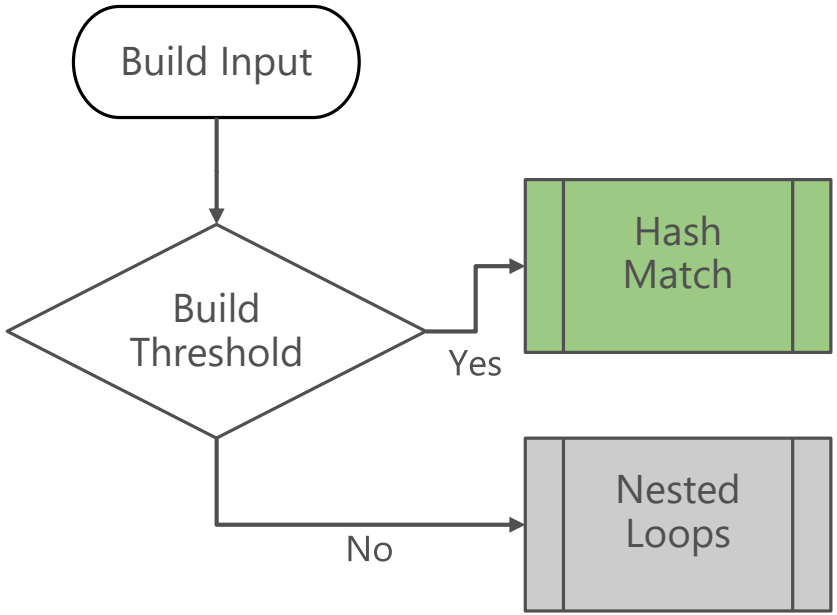
Batch Mode Adaptive Joins (AJ)

If cardinality estimates are skewed, we may choose an inappropriate join algorithm.

AJ will defer the choice of Hash Match or Nested Loops join until after the first join input has been scanned.

Adaptive Buffer is used up to the point where it's needed as the Build Table for HJ, or Outer Table for NLJ – Threshold is dynamic

AJ uses Nested Loops for small inputs, Hash Match for large inputs.



Demo

Adaptive Joins

SQL intersection

Up to 140 database compatibility level

Table Variable Deferred Compilation

Legacy behavior

Area	Temporary Tables	Table Variables
Manual stats creation and update	Yes	No
Indexes	Yes	Only inline index definitions allowed.
Constraints	Yes	Only PK, uniqueness and check constraints.
Automatic stats creation	Yes	No
Creating and using a temporary object in a single batch	Compilation of a statement that references a temp table that doesn't exist is deferred until the first execution of the statement	A statement that references a table variable is compiled along with all other statements before any statement that populates the TV is executed, so compilation sees it as "1".

150 database compatibility level

Table Variable Deferred Compilation

Azure SQL Database and SQL Server 2019 behavior

Area	Temporary Tables	Table Variables
Manual stats creation / update	Yes	No
Indexes	Yes	Only inline index definitions allowed.
Constraints	Yes	Only PK, uniqueness and check constraints.
Automatic stats creation	Yes	No
Creating and using a temporary object in a single batch	Compilation of a statement that references a temp table that doesn't exist is deferred until the first execution of the statement	Compilation of a statement that references a table variable that doesn't exist is deferred until the first execution of the statement

Demo

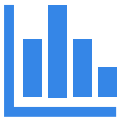
Table Variable Deferred Compilation

SQL intersection

APPROX_COUNT_DISTINCT
When approximate is good enough…

Provides approximate COUNT DISTINCT for big data scenarios with the benefit of high performance and a **(very) low memory** footprint.

Dashboard scenarios and trend analysis against big data sets with many distinct values (for example, distinct orders counts over a time period) – and many concurrent users where exact values are not necessary.



Data science big data set exploration. Need to understand data distributions quickly and exact values are not paramount.

Not banking applications or anywhere an exact value is required!

Demo

Approximate QP

T-SQL Scalar User-Defined Functions (UDFs)

User-Defined Functions that are implemented in Transact-SQL and return a single data value are referred to as **T-SQL Scalar User-Defined Functions**

T-SQL UDFs are an elegant way to achieve code reuse and modularity across SQL queries

Some computations (such as complex business rules) are easier to express in imperative UDF form

UDFs help in building up complex logic without requiring expertise in writing complex SQL queries

T-SQL Scalar UDF performance issues!

Iterative invocation: Invoked once per qualifying row. Repeated context switching – and even worse for UDFs that have T-SQL queries that access data

Lack of costing: Scalar operators are not costed (realistically)

Interpreted execution: Each statement itself is compiled, and the compiled plan is cached. Although this caching strategy saves some time as it avoids recompilations, each statement executes in isolation. No cross-statement optimizations are carried out.

Serial execution: SQL Server does not allow intra-query parallelism in queries that invoke Scalar UDFs. In other words, Scalar UDFs are parallelism inhibitors.

150 database compatibility level

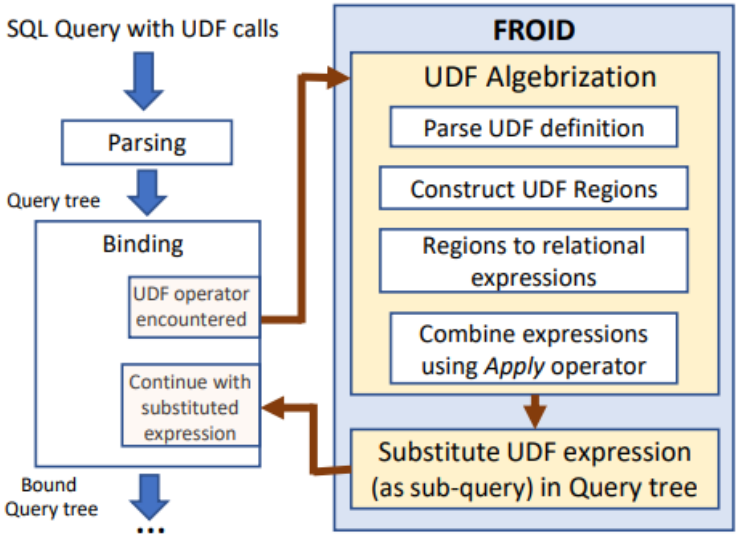
T-SQL Scalar UDF Inlining

Enable the benefits of UDFs without the performance penalty!

- Goal of the Scalar UDF Inlining feature is to improve performance for queries that invoke scalar UDFs where UDF execution is the main bottleneck

With SQL 2019/DB Compat 150:

- Using query rewriting techniques, UDFs are transformed into equivalent relational expressions that are "inlined" into the calling query



Source: "Froid: Optimization of Imperative Programs in a Relational Database"

T-SQL Scalar UDF Inlining

Table 1: Relational algebraic expressions for imperative statements (using standard T-SQL notation from [33])

Imperative Statement (T-SQL)	Relational expression (T-SQL)
DECLARE {@*var* *data_type* [= *expr*]}[, . . . *n*];	SELECT {*expr*|*null* AS *var*}[, . . . *n*];
SET {@*var* = *expr*}[, . . . *n*];	SELECT {*expr* AS *var*}[, . . . *n*];
SELECT {@*var*1 = *prj_expr*1}[, . . . *n*] FROM *sql_expr*;	{SELECT *prj_expr*1 AS *var*1 FROM *sql_expr*}; [, . . . *n*]
IF (*pred_expr*) {*t_stmt*; [. . . *n*]} ELSE {*f_stmt*; [, . . . *n*]}	SELECT CASE WHEN *pred_expr* THEN 1 ELSE 0 END AS *pred_val*; {SELECT CASE WHEN *pred_val* = 1 THEN *t_stmt* ELSE *f_stmt*; }[. . . *n*]
RETURN expr;	SELECT *expr* AS *returnVal*;

Source: "Froid: Optimization of Imperative Programs in a Relational Database"

Scalar UDF Inlining non-starters

Non-inlineable constructs:

- Invoking any intrinsic function that is either time-dependent (such as GETDATE()) or has side effects (such as NEWSEQUENTIALID())
- Referencing table variables or table-valued parameters
- Referencing scalar UDF call in its GROUP BY clause
- Natively compiled (interop is supported)
- Used in a computed column or a check constraint definition – we've received lots of feedback on this scenario for this one
- References user-defined types
- Used in a partition function

To inline, or not to inline

See sys.sql_modules catalog view includes a property called is_inlineable:

- 1 indicates that it is inlineable, and 0 indicates otherwise
- Value of 1 for all inline TVFs as well

If a scalar UDF is inlineable, it doesn't imply that it will always be inlined. SQL Server will decide (on a per-query, per-UDF basis) whether to inline a UDF or not if:

- UDF definition has thousands of lines of code (itself or by using nesting)
- UDF used in a GROUP BY clause

Decision is made when the query referencing a scalar UDF is compiled.

Mediterranean Shipping Company

"With the combination of Intelligent Query Processing in SQL Server 2019 as the second generation of Adaptive Query Processing as well as Memory-Optimized TempDB Metadata, we are now able to achieve incredible performance in a more predictable way for all our business-critical processes. We are really happy with SQL Server 2019; key features such as scalar UDF Inlining and Table Variable Deferred Compilation enhance our developers' productivity and empower MSC in providing improved solutions for our customers."

Javier Villegas, Global Database Administrator & Design Coordinator
Mediterranean Shipping Company

msc
MEDITERRANEAN SHIPPING COMPANY

Demo

Scalar UDF Inlining

SQL intersection

What's next?

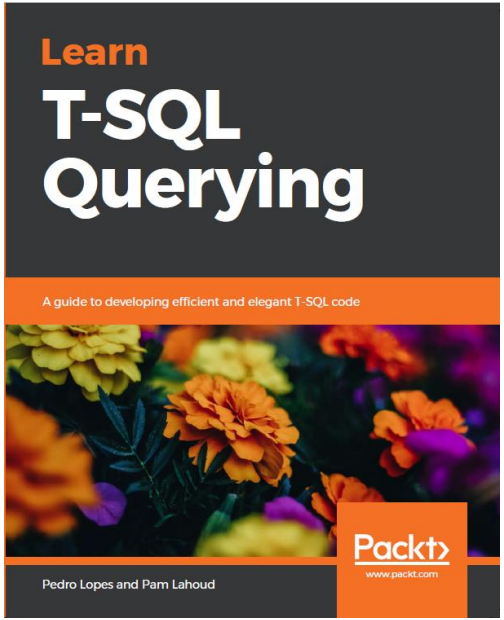
The features we saw today are in SQL Server 2019 and Azure SQL Database*

We continue working on intelligent query processing features as we speak – share your scenarios with us!

Please email IntelligentQP@microsoft.com

Learn more

Download and try SQL Server 2019	https://aka.ms/ss19
Check out these great data-related demos	https://aka.ms/DataSamples https://aka.ms/IQPDemos https://aka.ms/SQL2019Notebooks
Continue learning with our new book	https://aka.ms/LearnTSQLQuerying https://aka.ms/LearnTSQLQuerying_errata
One shortcut to rule them all!	https://aka.ms/SQLShortcuts



Questions?

Microsoft

Don’t forget to complete an online evaluation!

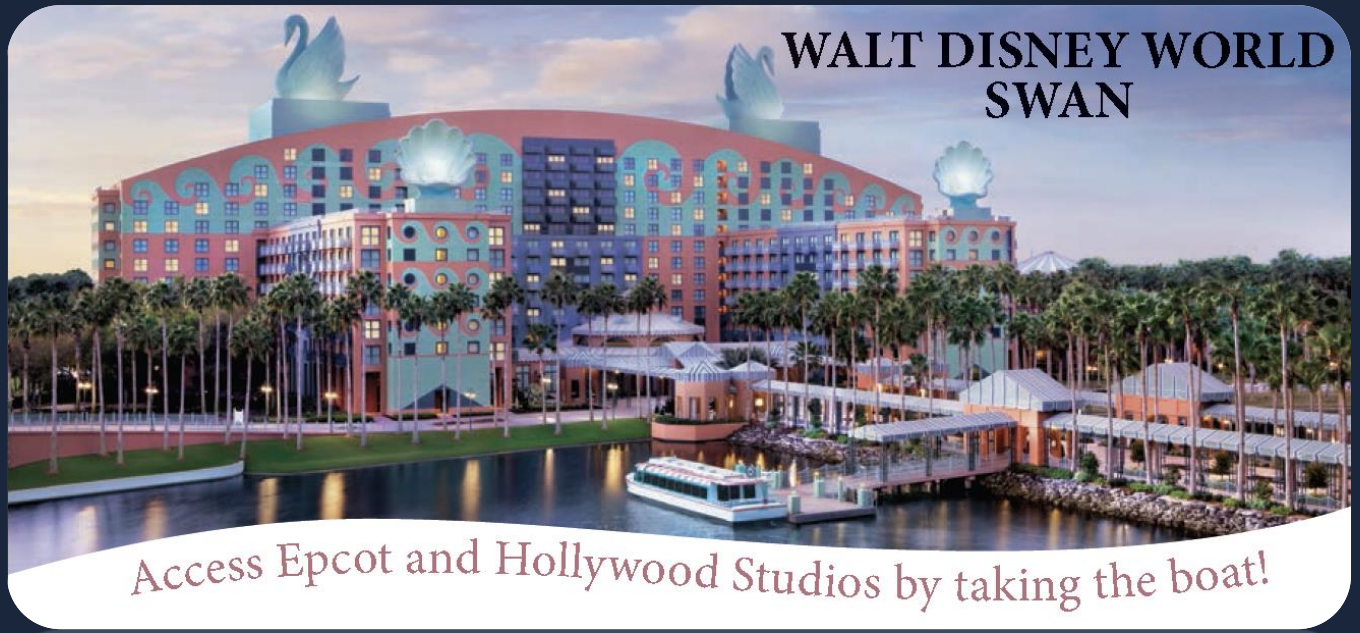
SQL Server’s Path Toward an Intelligent Database

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

Thank you!

Save the Date!

www.SQLintersection.com



2020

Week of April 6

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!