Indexing Basics

Speaker: Paul Tuohy

High Performance, Resilient APIs for Your IBM i Using Kafka

Speaker: Dan Magid

Paul's handout at
https://cutt.ly/IndexingBasics
or
https://cutt.ly/IndexingBasics_SiD

Sponsored by

dan@eradani.com

See more Summit Lunch & Learn webinars at
SystemiDeveloper.com/LunchLearn

Indexing Basics

Paul Tuohy
ComCon
System *i* Developer
5, Oakton Court,
Ballybrack
Co. Dublin
Ireland

Phone: +353 1 282 6230
e-Mail: paul@systemideveloper.com
Web: www.systemideveloper.com
www.ComConAdvisor.com

IBM CHAMPION

System *i* Developer
The Summit of Education Excellence

ComCon

Paul Tuohy

System i Developer
The Summit of Education Excellence
ComCon

Paul Tuohy, author of "*Re-engineering RPG Legacy Applications*" and "*The Programmer's Guide to iSeries Navigator*", is one of the most prominent consultants and trainer/educators for application modernization and development technologies on the IBM Midrange. He currently holds positions as CEO of ComCon, a consultancy firm based in Dublin, Ireland, and founding partner of System i Developer, the consortium of top educators who produced the acclaimed *RPG & DB2 Summit* conference. Previously, he worked as IT Manager for Kodak Ireland Ltd. and Technical Director of Precision Software Ltd.

In addition to hosting and speaking at the RPG & DB2 Summit, Paul is an award-winning speaker at COMMON, COMMON Europe Congress and other conferences throughout the world. His articles frequently appear in iProDeveloper, The Four Hundred Guru, RPG Developer and other leading publications. Paul also hosts the popular *iTalk with Tuohy* podcast interviews.

This presentation may contain small code examples that are furnished as simple examples to provide an illustration. These examples have not been thoroughly tested under all conditions. We therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All code examples contained herein are provided to you "as is". THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Agenda

Indexing on IBM i

Indexes in action

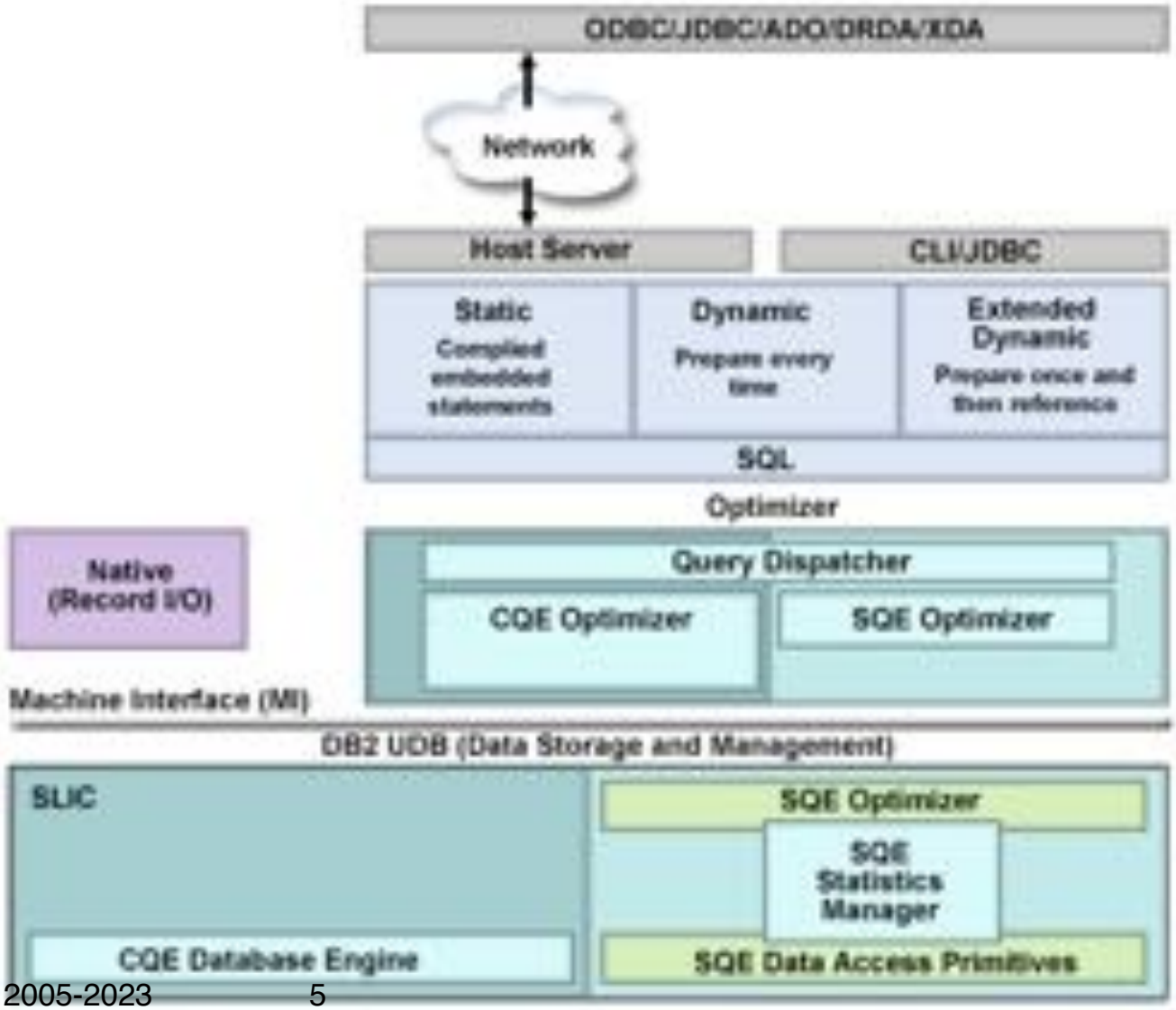
Optimization

Identify the rows of interest while avoiding useless data

Optimizer makes use of access paths

Access paths come from

- Key constraints
- Keyed physical files
- Keyed logical files
- Indexes



Indexes

Binary Radix

- Traditional format
- A multilevel hybrid tree structure

Encoded Vector Index (EVI)

- Unique to IBM i
- A variation on bitmap indexing


```
create index forindex02 on forindex (somevalue desc, keyid );
create index forindex06 on forindex (keyid, somevalue) where somevalue <= 10000;
create encoded vector index forindex08 on forindex
         (somecode) include (count(*), sum(somevalue));
```

Index Specifics

Index by column value/sequence

- Order by

Derived Key Index

- Index is a derived value of one or more columns
- Result of an expressions
- Result of a scalar function

Sparse Indexes

- Content of index based on a WHERE clause

Index only access

- All required column values are in an index
- Aggregate values maintained in an EVI

Indexes are NOT

- Specifically for unique keys
  - That is what constraints are for
- Just for sequence

Index Uses

System i Developer
The Summit of Education Excellence
ComCon

Native I/O

Identify and process rows in a table

- Selection
- Joining
- Grouping
- Ordering

Statistical Information

- Number of distinct values
- Distribution of values

Index Strategies

Proactive and Reactive

Proactive

- Anticipate column requirements
  - for selection, joining, grouping, and ordering
- Build indexes based on requirements
  - Views can provide a good indication of column requirements

Reactive

- Build indexes based on
  Optimizer feedback, Query implementation plan, System performance measurements

A reactive approach will be required - how reactive depends on how proactive you were

The optimizer advises indexes that would be useful

- But not all possible useful indexes are advised
- Not all advised indexes are required

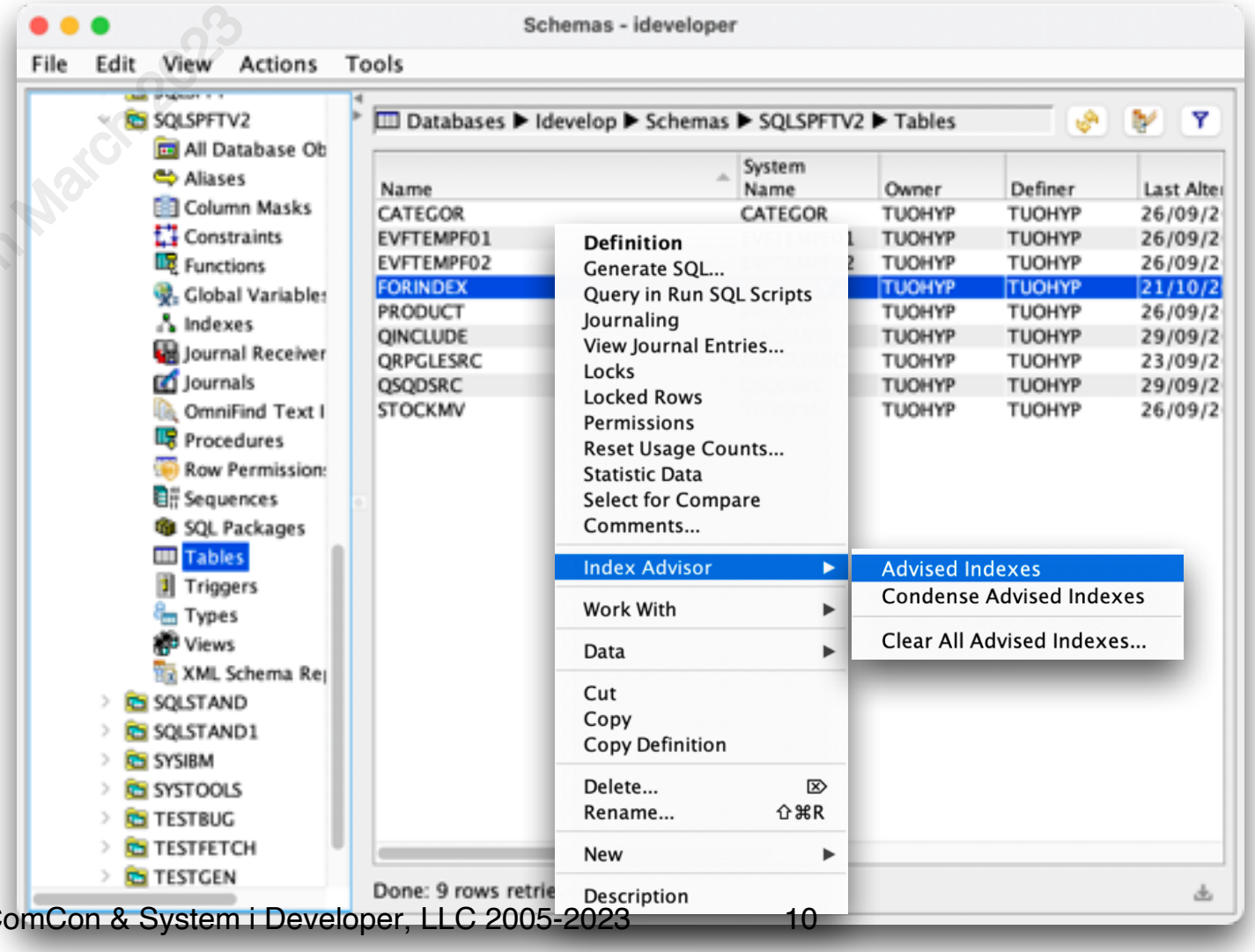
Index advise is in the catalog table SYSIXADV

index Advisor

System i Developer
The Summit of Education Excellence
ComCon

Index advise is available for a table, schema or system

- In ACS schemas - right click on table, schema or system name and select Index Advisor
- Index advise also available for SQL statements
  - In Visual Explain - menu option *Actions>Index and Statistics Advisor*

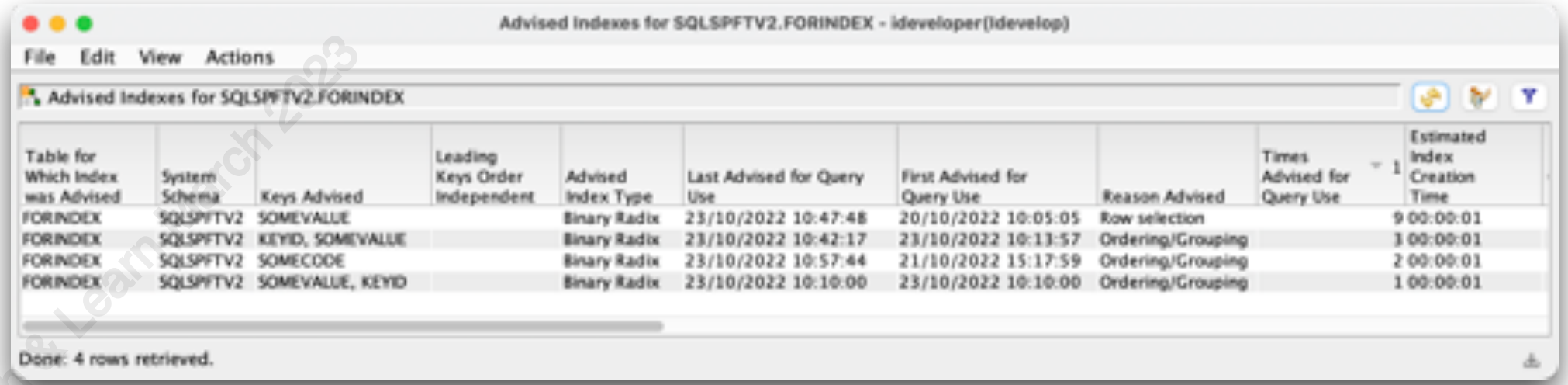


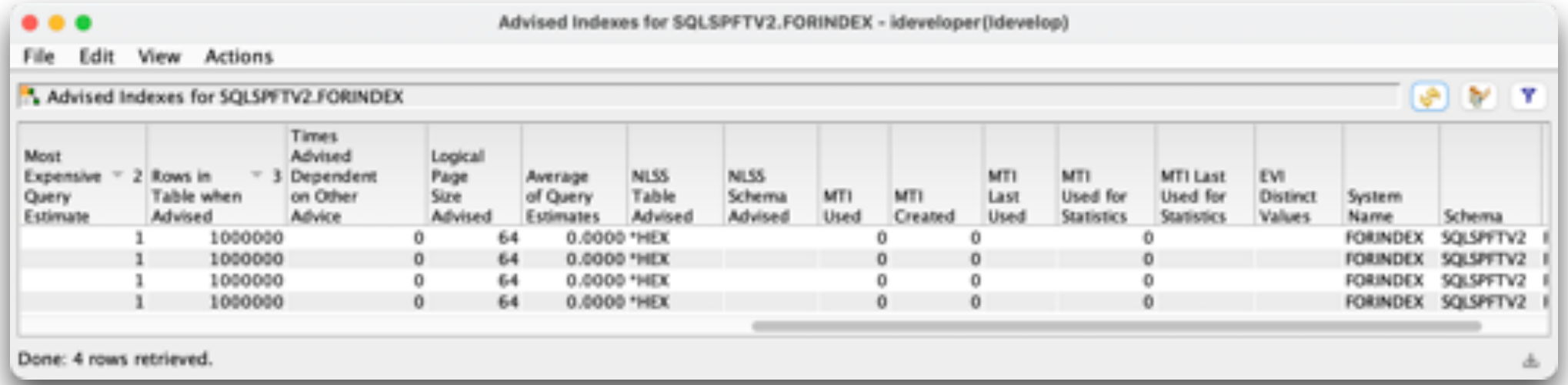
Index Advisor

System i Developer
The Summit of Education Excellence
ComCon

This is a customized view

- *View>Columns*





Index Advisor Information

The advised key

The first and last time the index was advised

The number of times the index was advised

The longest and average execution times for the queries that generated the index advise

The estimated index creation time

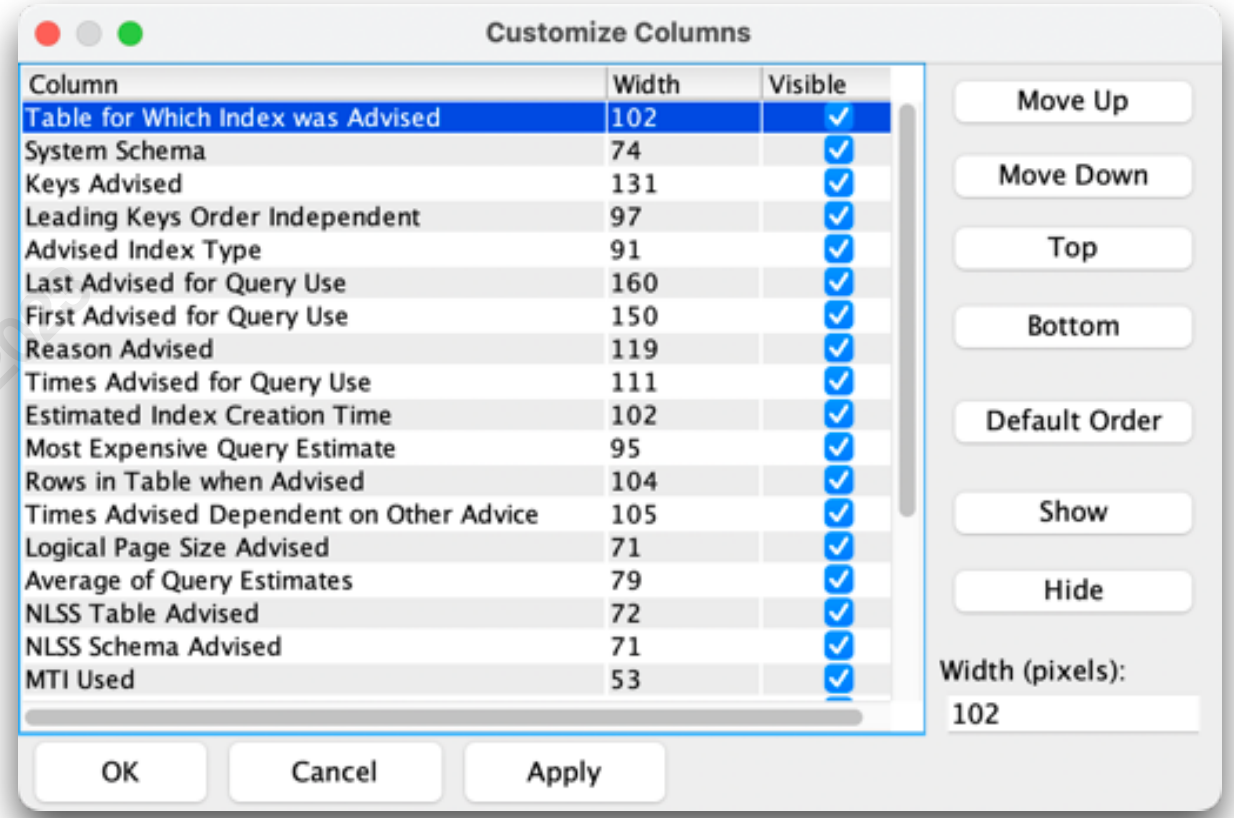
The number of rows in the table the last time the index was advised

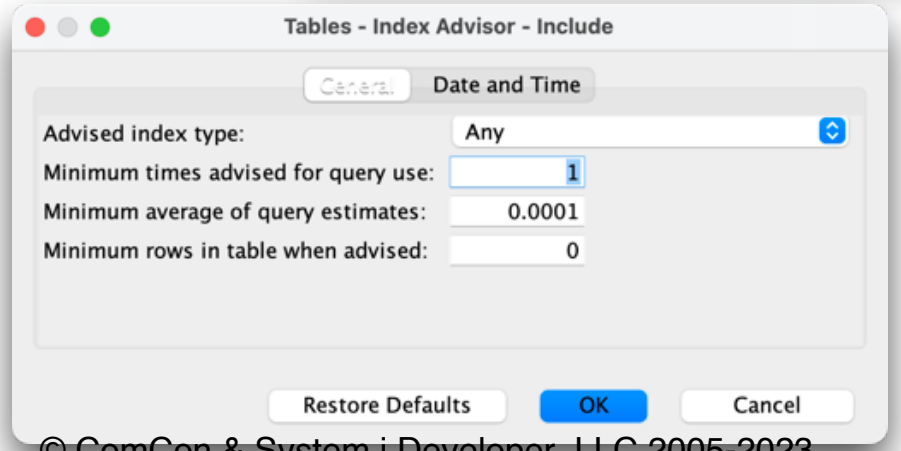
The reason the index was advised (row selection and/or ordering/grouping)

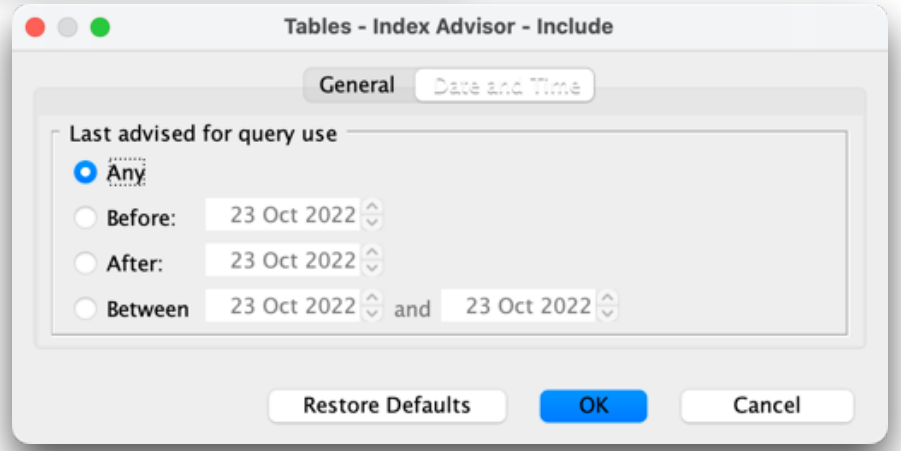
Condensed Advised Indexes

- Condenses multiple like advised indexes to one

Index Advisor - View Options



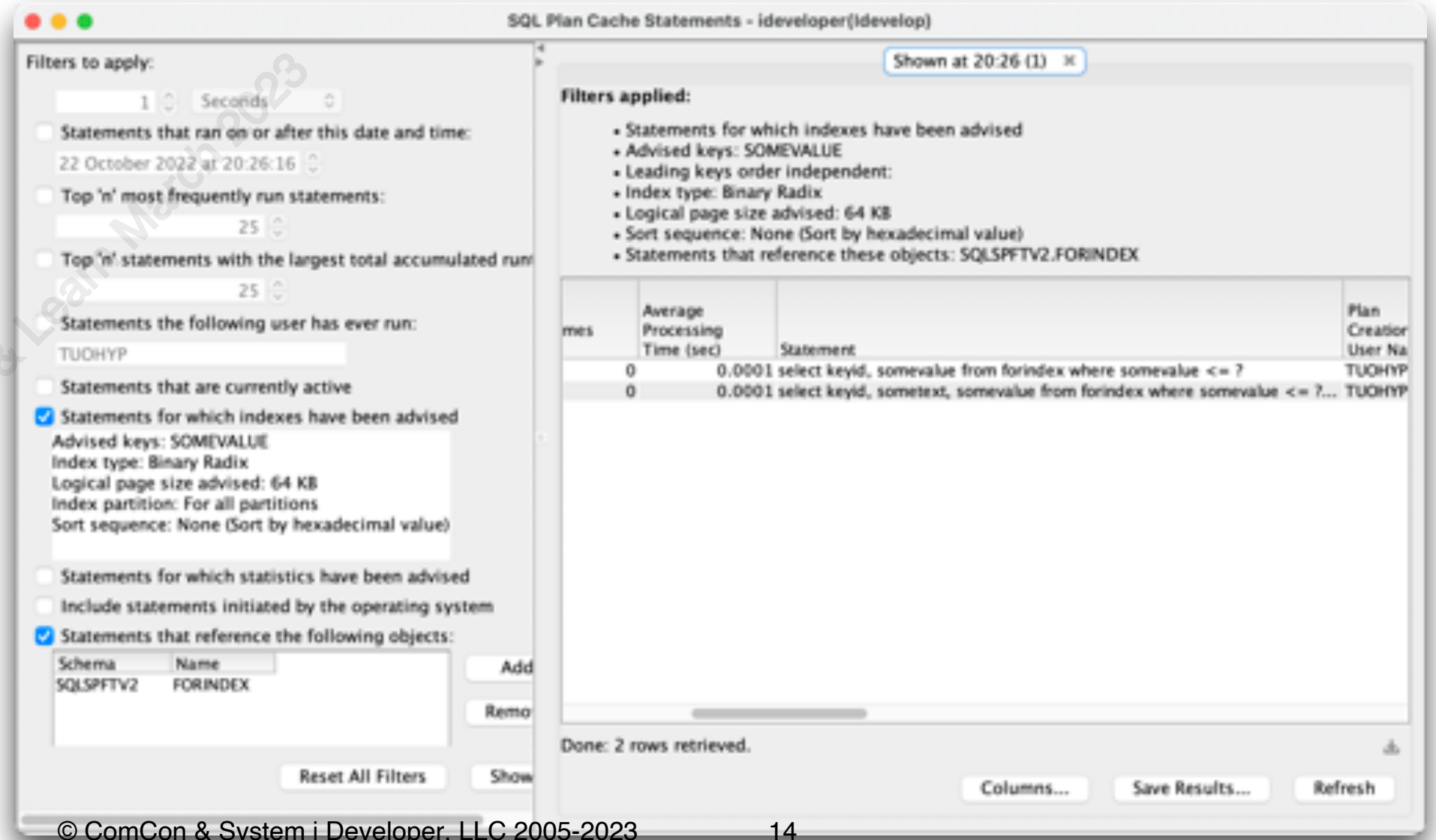




Index Advisor - Actions>Work With

SQL Plan Cache Statements

- If still available



Db2 for i Services

Services available to help with Index Maintenance

ACT_ON_INDEX_ADVICE procedure

- Create new indexes for a table based on Index advise

REMOVE_INDEXES procedure

- Drop any indexes meeting specified criteria

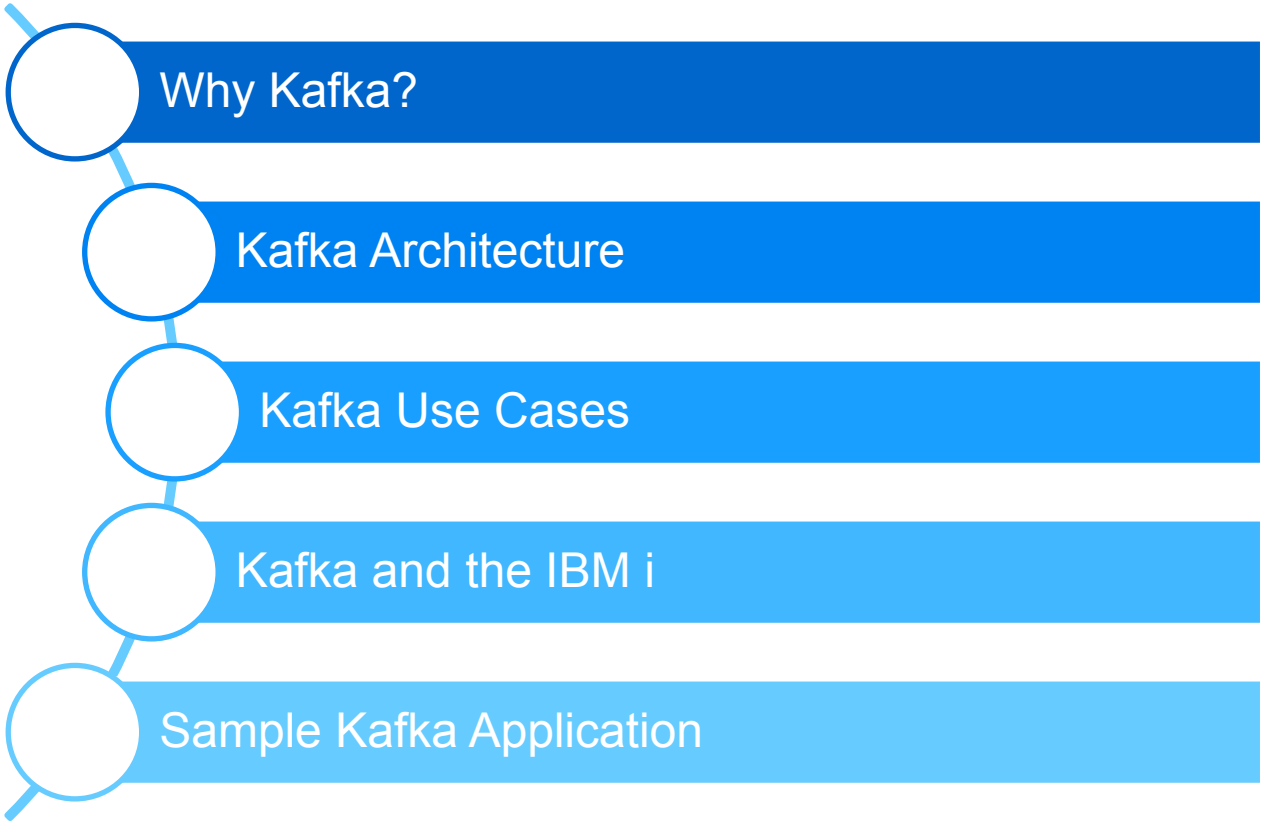
RESET_TABLE_INDEX_STATISTICS procedure

- Clears usage statistics for indexes defined over tables
  - Optionally delete rows from the index advice table

High Performance, Resilient APIs for Your IBM i Using Kafka

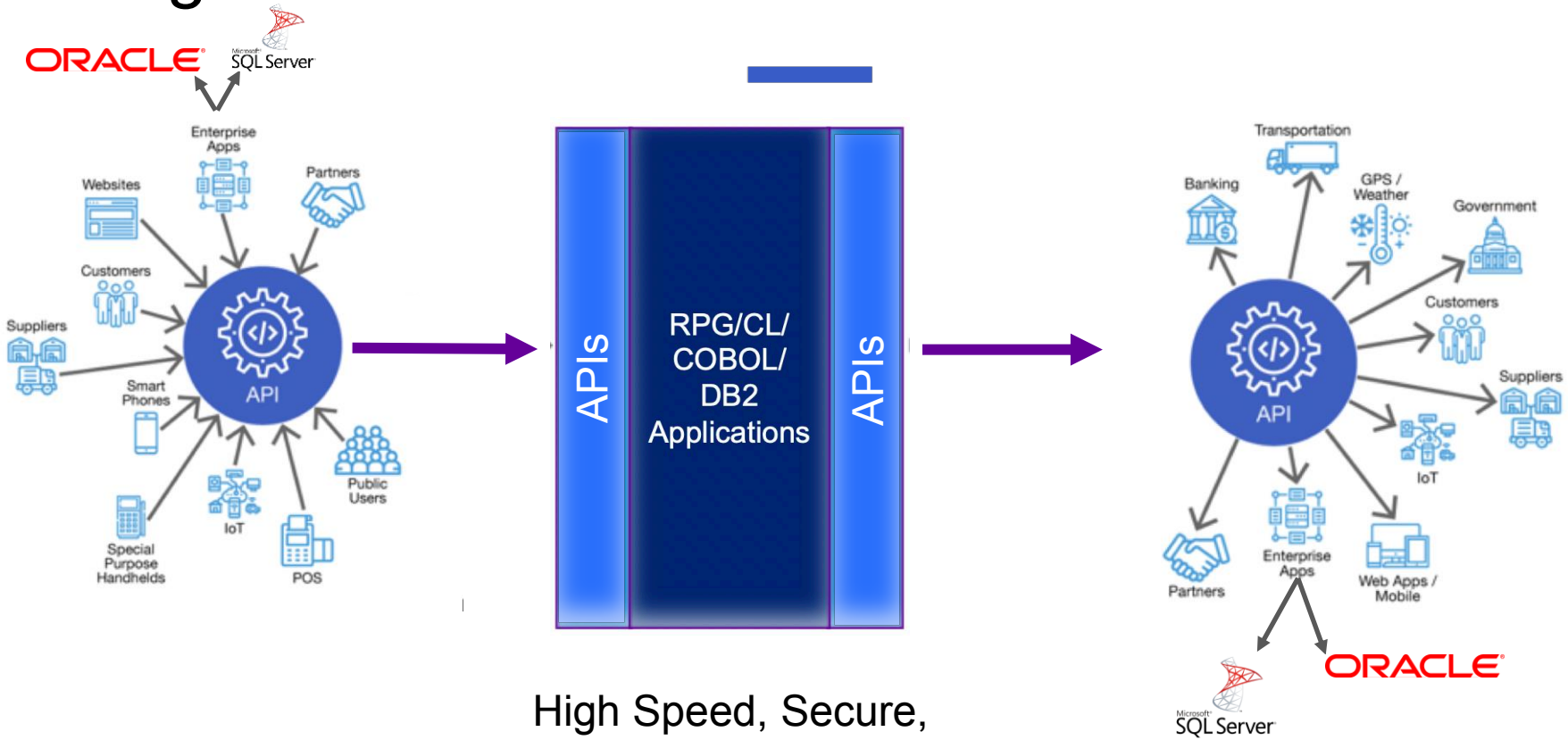
Eradani

System i Developer RPG & DB2 Summit Lunch & Learn, Spring 2023



Eradani

High Performance API Connections and IBM i



High Speed, Secure,
Easy-to Maintain
& Flexible Connections

REST Services vs Apache Kafka

REST

- Call to URL using standard HTTP Methods (GET, PUT, Patch, Delete, etc.)
- Easy to setup and use
- Low latency for small payloads and relatively low call volumes
- Highly flexible connections
- Slow for high volume calls and large message payloads

Apache Kafka

- High speed processing for large volume message streams
- Real time processing (replace batch & ETL processes)
- Handles large data messages
- Automated resilience
- Pub/Sub, Event Driven architecture
- More complex to setup and use

“Apache Kafka is a high-performance, resilient, open source-based, event-driven, streaming, pub-sub messaging application for providing loosely coupled connections between a variety of message producers and consumers.”

Kafka Use is Exploding…

- Over One Billion Downloads
- Used by More Than 80% of the Fortune 500
- 300% Growth in Companies Using More Than 50 Kafka Clusters
- LinkedIn is Processing 7 Trillion Messages a Day via Kafka
- Facilitating the Move to Stream Processing Rather Than Batch

Eradani

Why Event Driven Architecture

Procedural:

Check Database of Calls on a Scheduled Basis:

“Is there an emergency?”

Event Driven:

“There is an Emergency!”

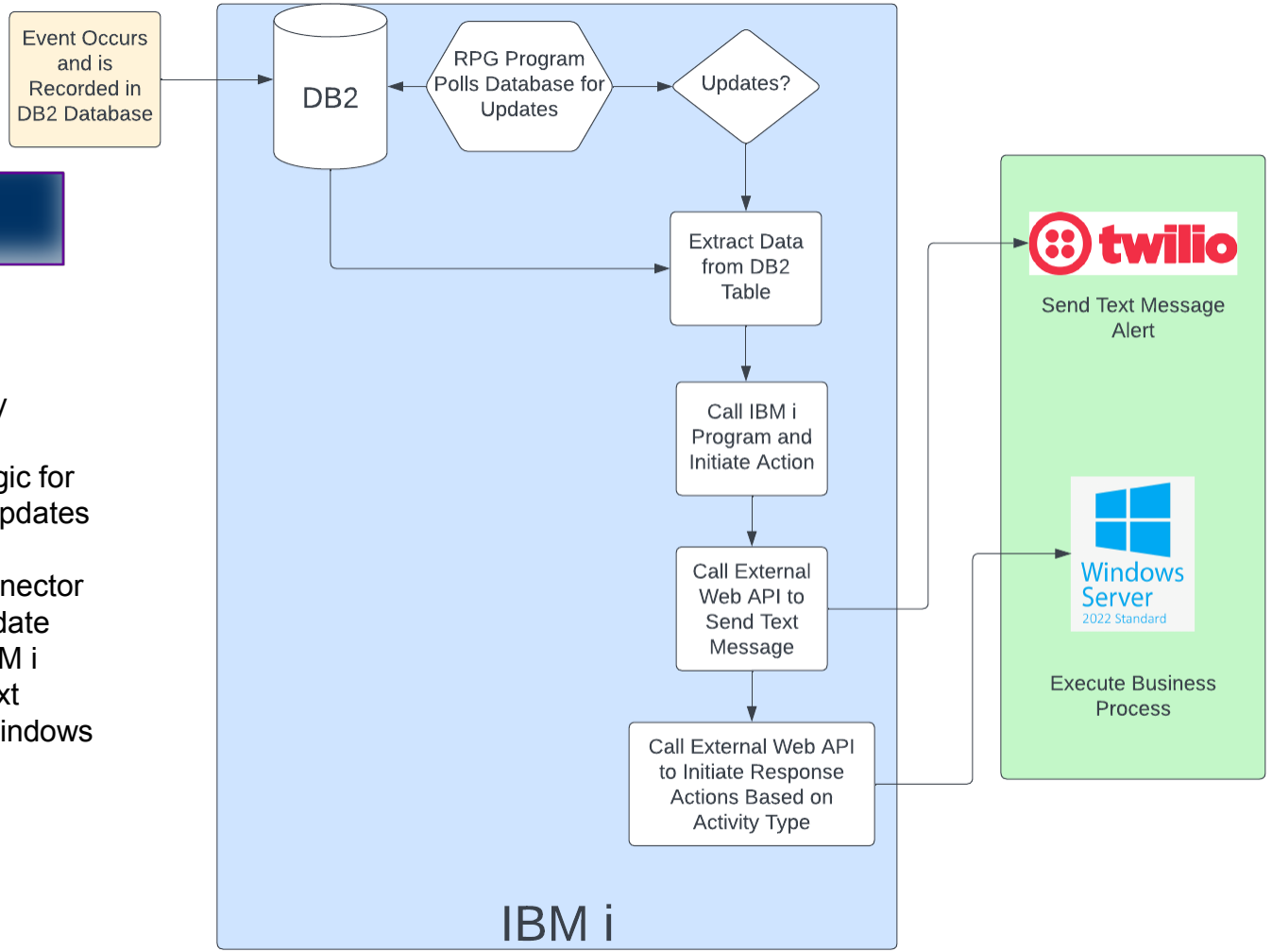
Event Driven:

- When Data Delivery is Continuous and/or Irregular and/or Unpredictable
- When Response is Potentially Required to Each Record Received
- When Response Must be Fast (Real time)

Polling Approach

Concerns:

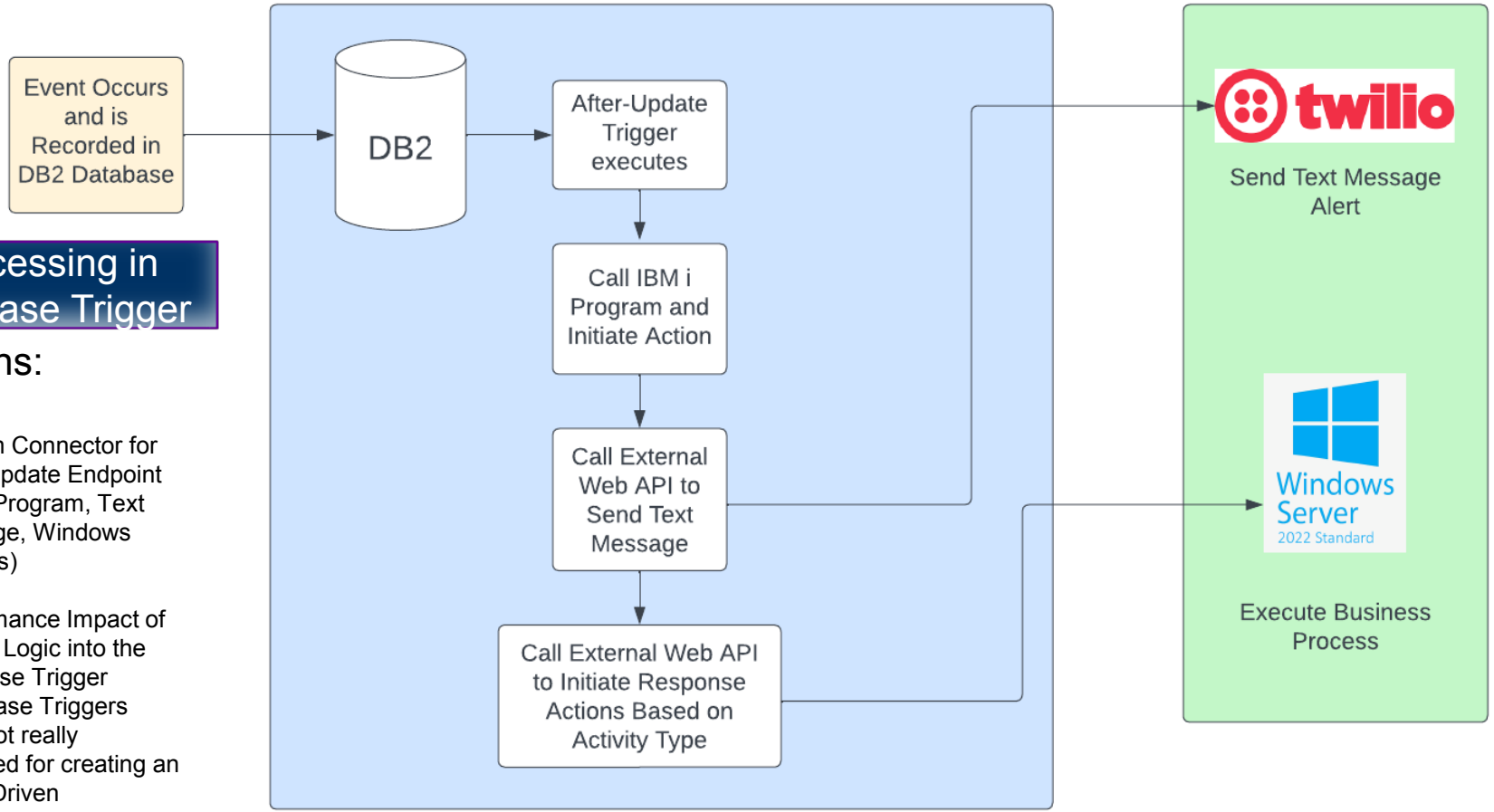
- Polling Delay
- Complex Logic for Identifying Updates
- Custom Connector for Each Update Endpoint (IBM i Program, Text Message, Windows Process)



Processing in Database Trigger

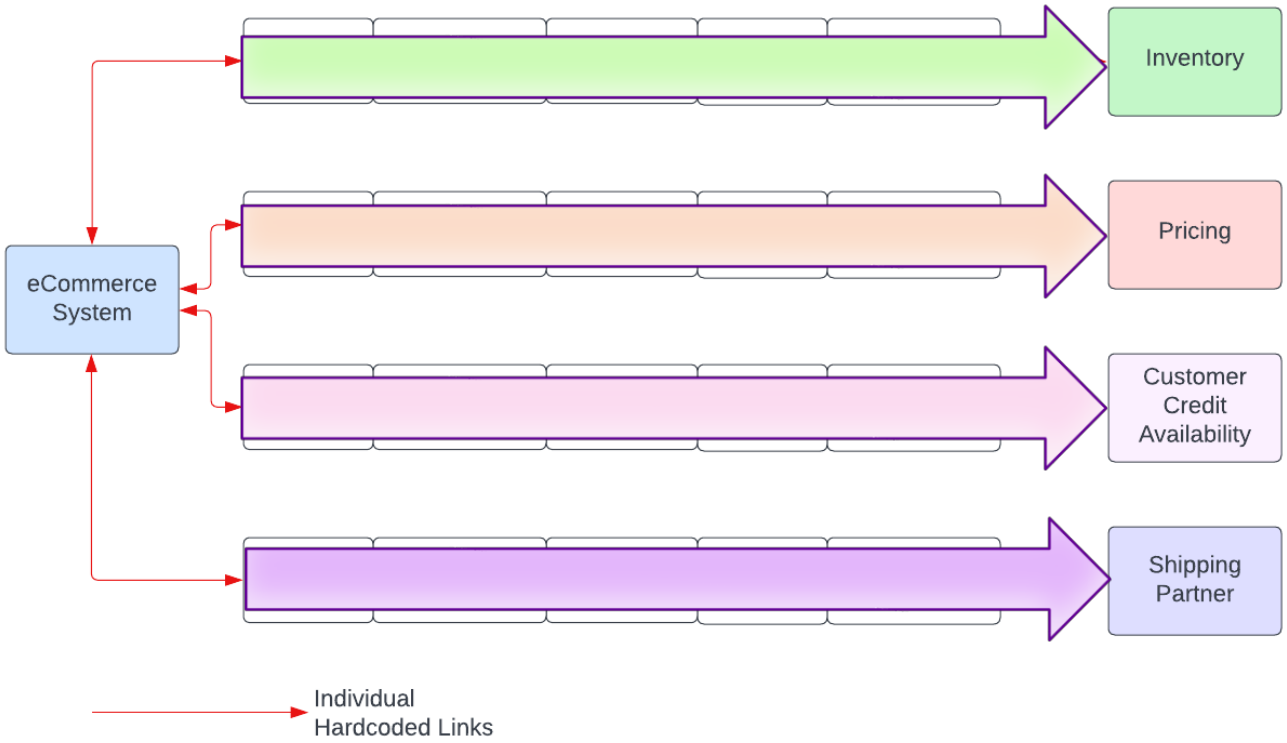
Concerns:

- Custom Connector for Each Update Endpoint (IBM i Program, Text Message, Windows Process)
- Performance Impact of Putting Logic into the Database Trigger (Database Triggers were not really designed for creating an Event Driven architecture)

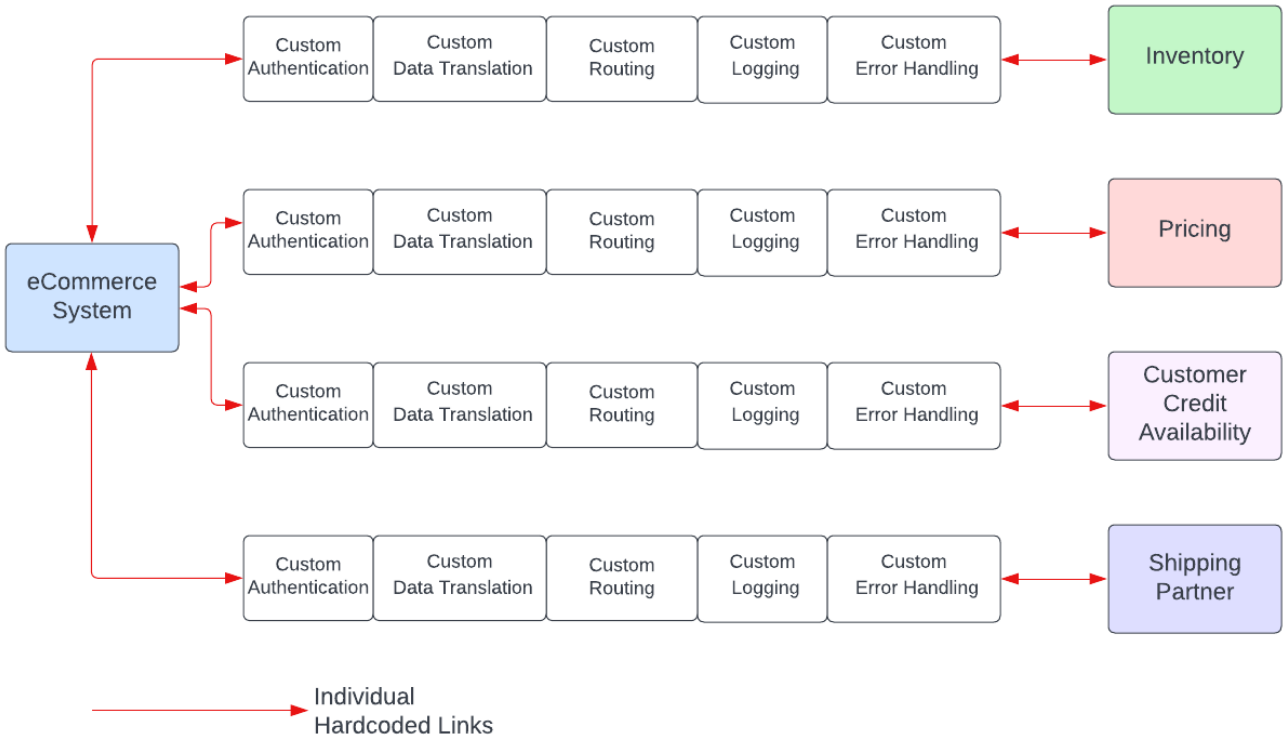


What is the Alternative?

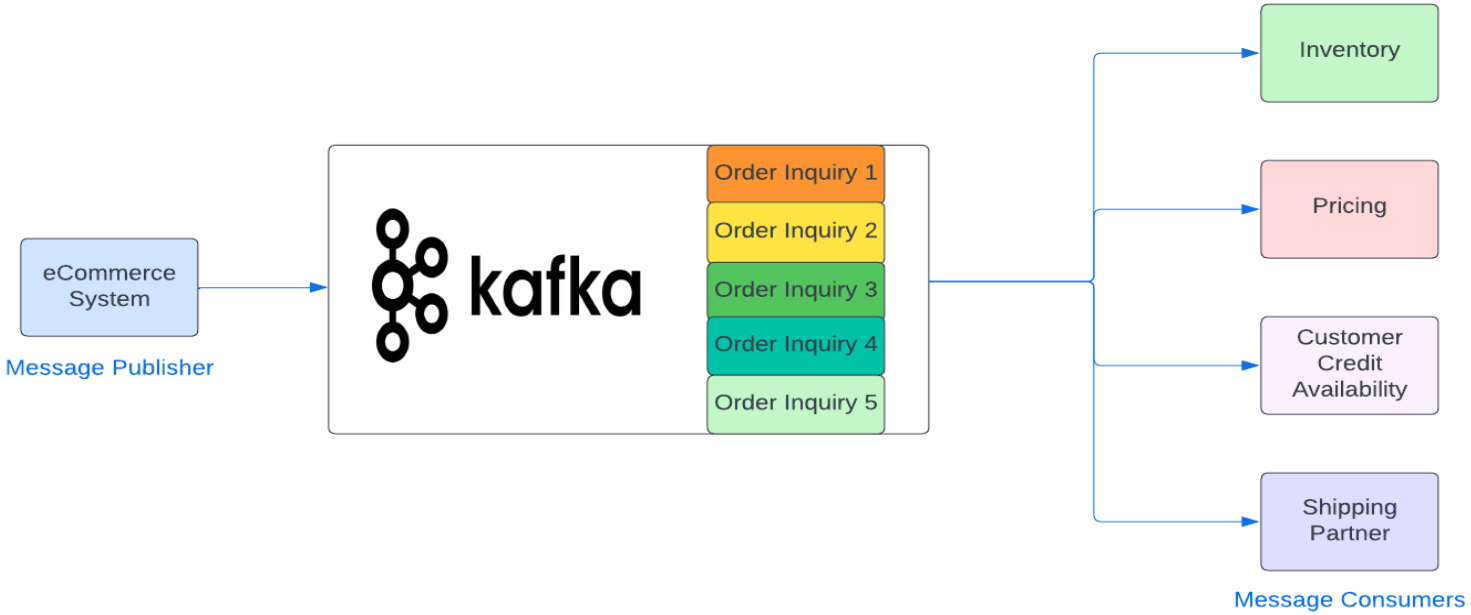
Events Require Integration Regardless of Approach



Direct Integration without Kafka



With Kafka Events are Posted Once



Producers

Arrival
Departure
Geo Coordinates
Fuel Stop

Deposit
Loan Payment
Withdrawal
Electronic Transfer

Blood Pressure
Oxygen Level
Pulse
Temperature

Producers

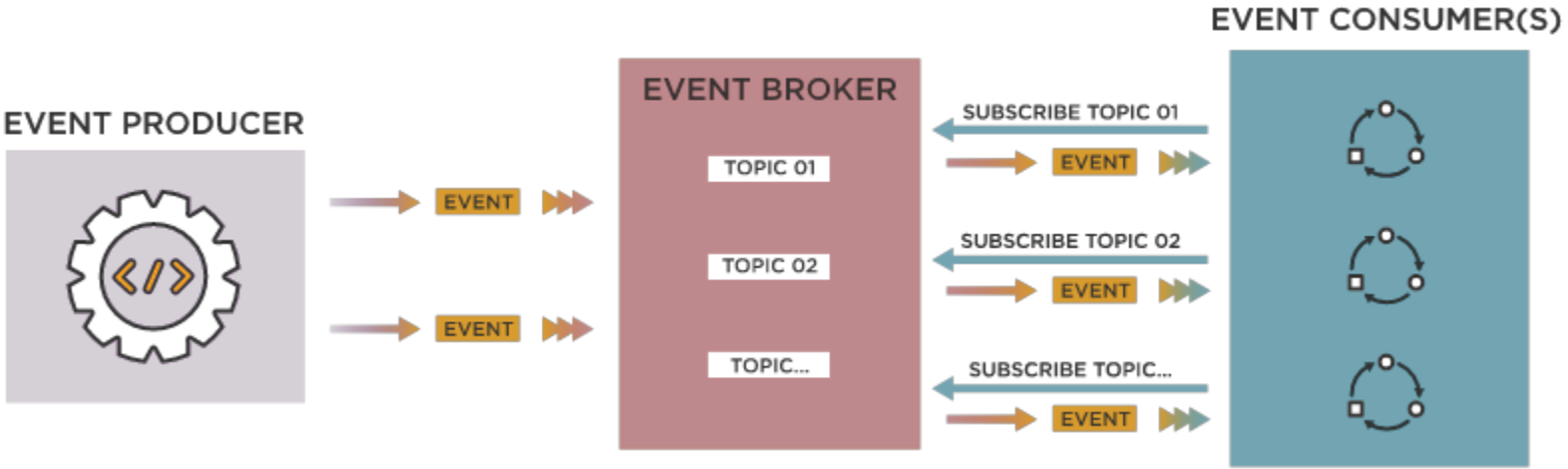
Inventory Inquiry
Request for Price
Order Entry

Item Location
Item Status
Required Inventory

Location Services
Ad Click
App Open

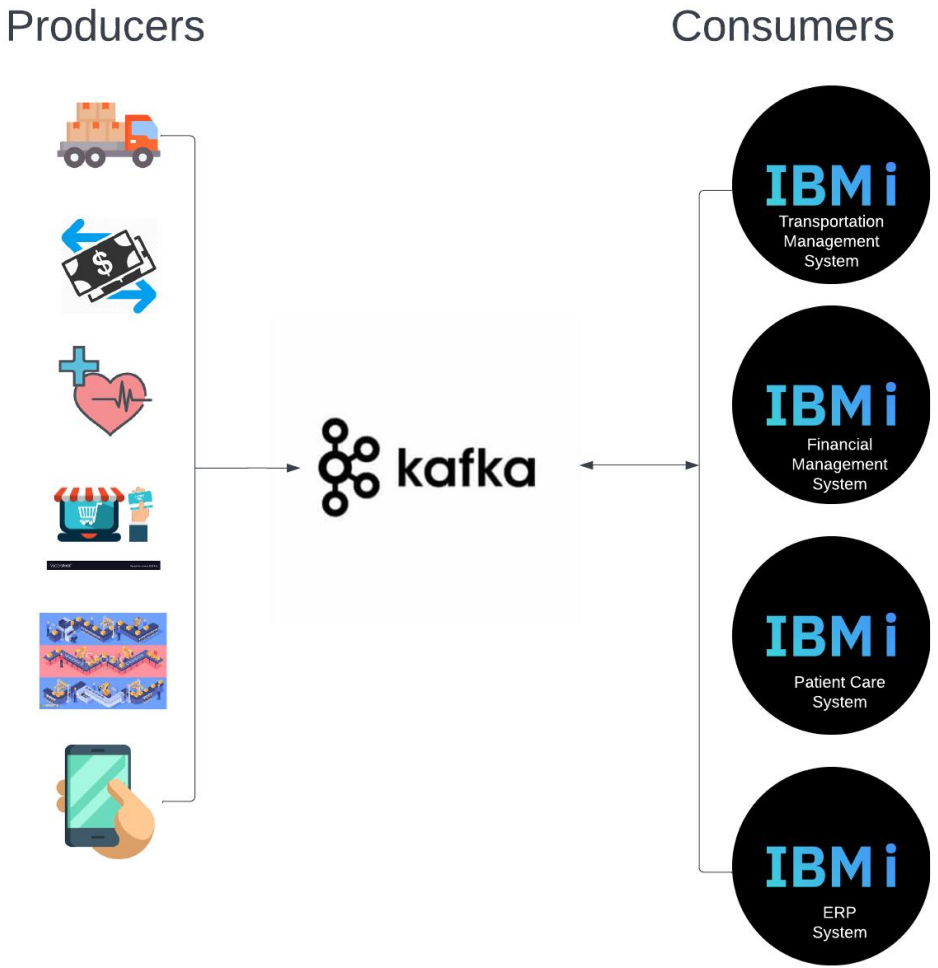
Eradani

Publish/Subscribe (Produce/Consume) Model

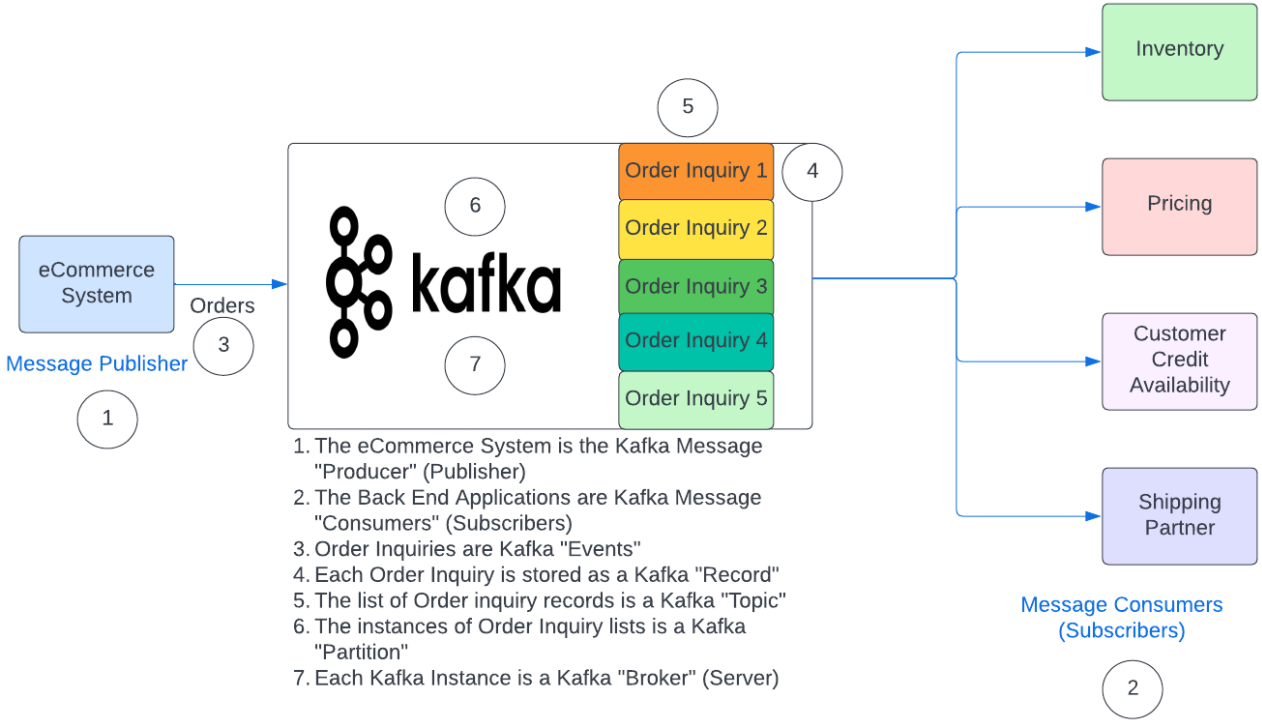


Decoupled Interface

- Producers are Unaware of Consumers
- Consumers are Unaware of Producers
- Add Consumers without Affecting Producers
- Consumer Performance Does Not Affect Producers



Pub/Sub Architecture with Kafka

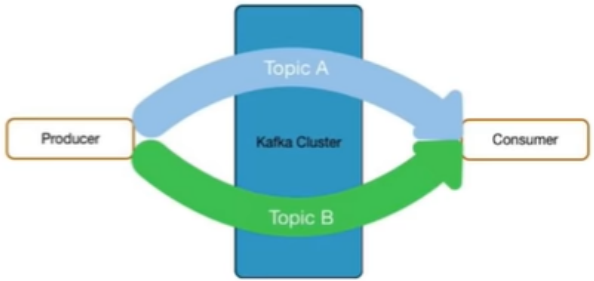


1. The eCommerce System is the Kafka Message "Producer" (Publisher)
2. The Back End Applications are Kafka Message "Consumers" (Subscribers)
3. Order Inquiries are Kafka "Events"
4. Each Order Inquiry is stored as a Kafka "Record"
5. The list of Order inquiry records is a Kafka "Topic"
6. The instances of Order Inquiry lists is a Kafka "Partition"
7. Each Kafka Instance is a Kafka "Broker" (Server)

Kafka Topics

Topics

- **Topics**: Streams of "related" Messages in Kafka
  - Is a **Logical Representation**
  - **Categorizes Messages** into Groups
- Developers define Topics
- Producer ←→ Topic: N to N Relation
- Unlimited Number of Topics



Kafka Topics

Blood Pressure Readings
3-3-2023, 04:00:00, 117, 72, 34567
3-3-2023, 06:00:00, 122, 81, 34567
3-3-2023, 08:00:00, 132, 83, 34567
3-3-2023, 10:00:00, 130, 80, 34567
3-3-2023, 12:00:00, 135, 78, 34567
3-3-2023, 14:00:00, 133, 81, 34567
3-3-2023, 16:00:00, 136, 79, 34567

Truck Arrivals & Departures
3-3-2023, 10:00:00, ARR, CHI, TR334
3-3-2023, 10:05:00, DEP, ATL, TR867
3-3-2023, 11:00:00, DEP, CLE, TR255
3-3-2023, 11:25:00, ARR, DET, TR129
3-3-2023, 12:00:00, DEP, CHI, TR334
3-3-2023, 14:00:00, ARR, BRM, TR867
3-3-2023, 21:00:00, ARR, CHI, TR255

Immutable log of related messages

Eradani

Replication for Resilience & High Performance

Subscribers Can Subscribe to Multiple Topics

Zookeeper

Kafka RAFT protocol

Manage the Clusters


Some Real World Kafka Use Cases
Eradani

Trucking Company
1M Rate Requests/Day

Distributor
1 Million+ Transactions
in FTP File

eCommerce
Query 80 Warehouses
With Sub 2 Second Response

Real-Time Patient
Monitoring

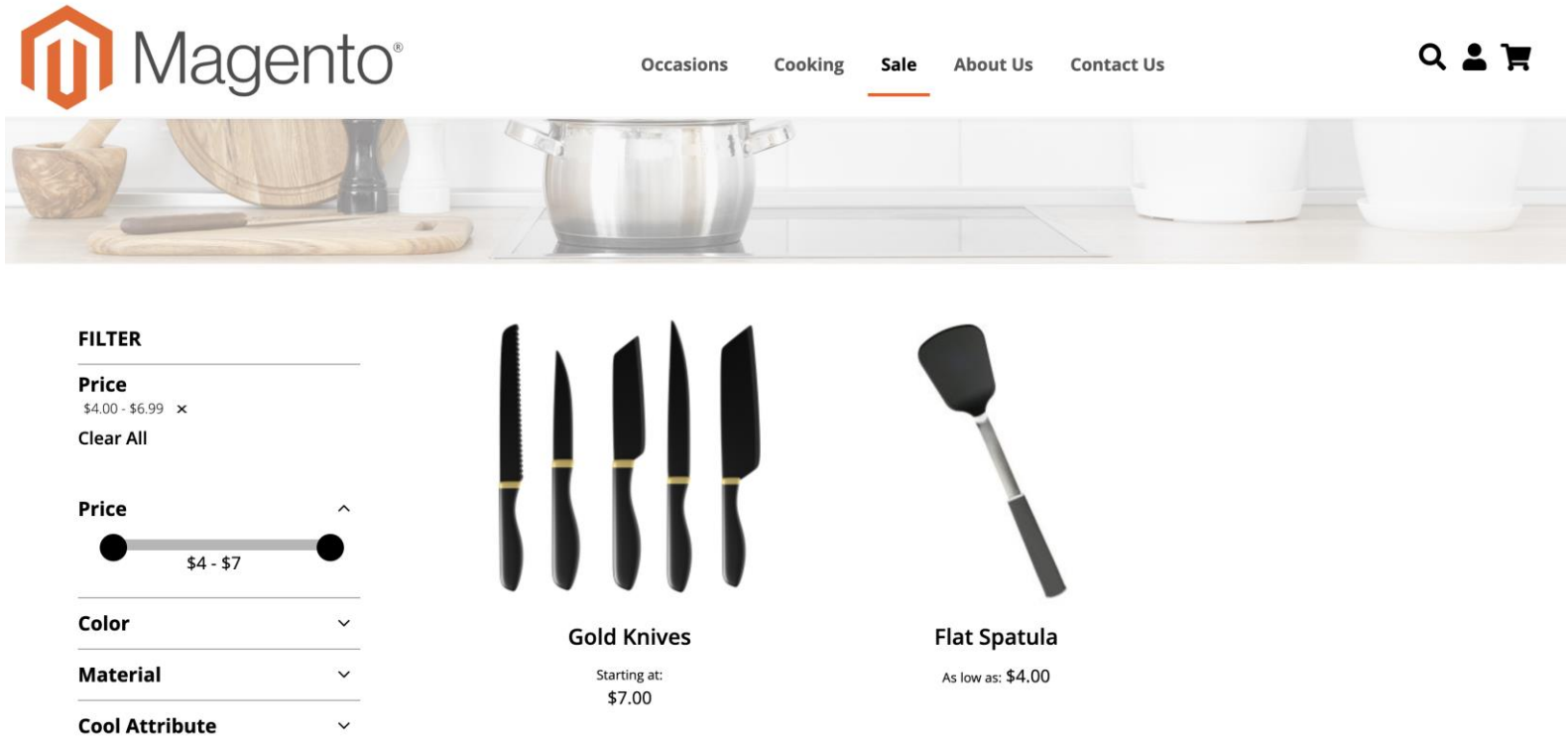
Financial Services
12,000 Transactions/Second

Retail Chain
Peak Load of
80,000 Transactions/Second

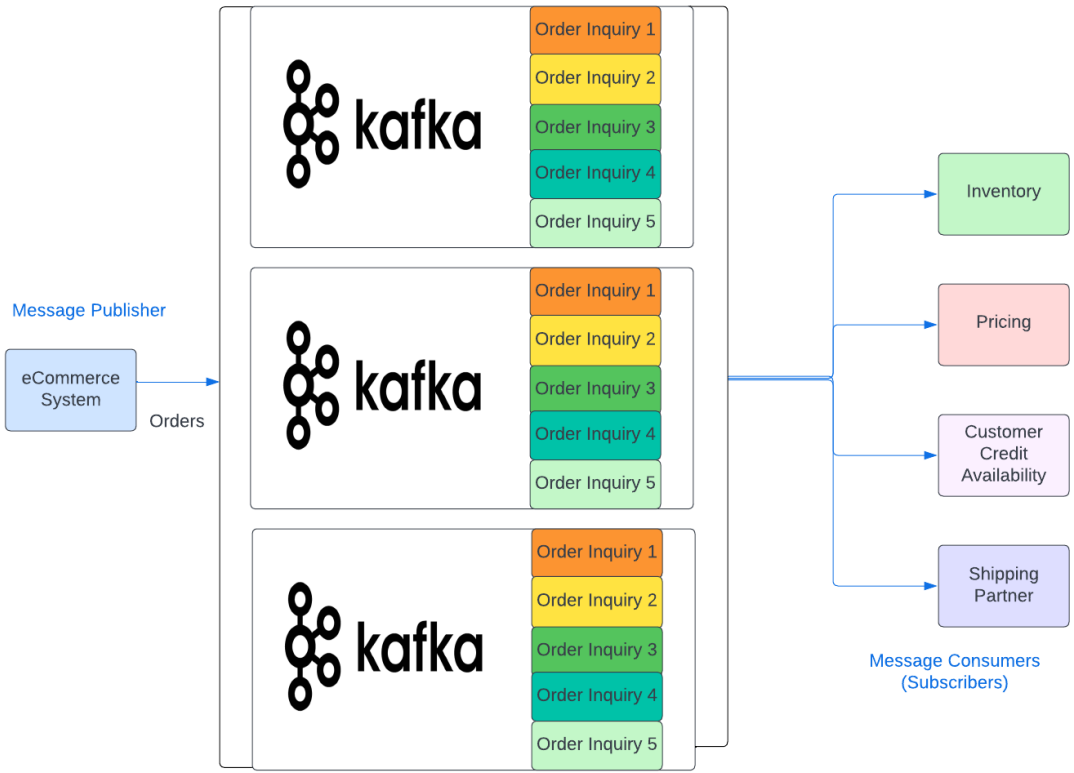
Insurance
Query Multiple Outside APIs
& Provide Instant Quotes

Eradani

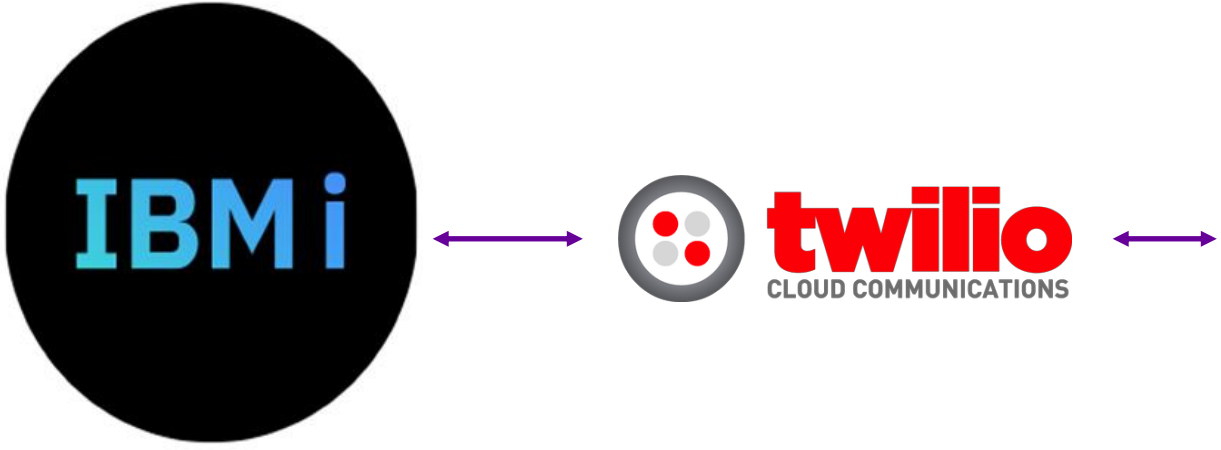
This is an IBM i Application - eCommerce

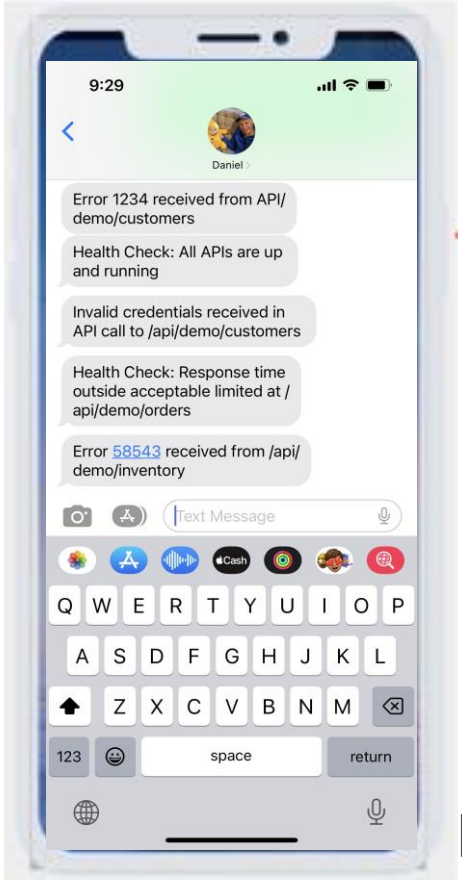


Resilience & Performance with Kafka Replication



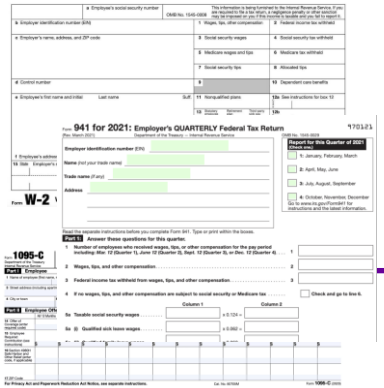
Add High Speed Inbound and Outbound Text Messaging to Any Application





Eradani

Open Source Cost Savings
Automate Business Processes



Integrate Form with IBM i Data

IBM i
RPG/COBOL/CL/DB2

Generate Documents

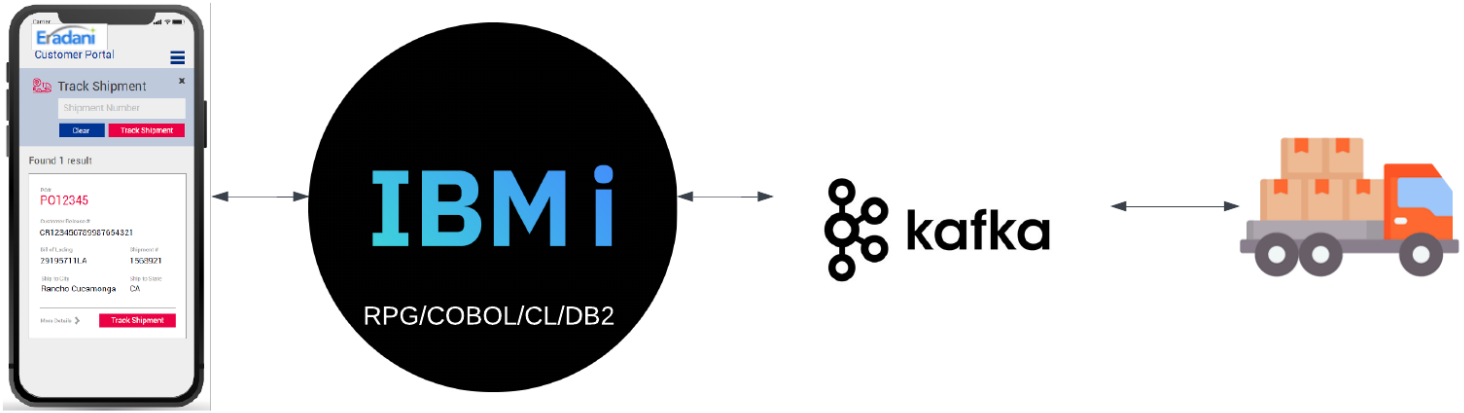
Post Documents to Kafka

kafka

Kafka can manage very large message payloads
The Message can include the document or simply a pointer to the document

Eradani

Real Time Vehicle Tracking



Sample Code in JavaScript for Sending Kafka Messages

```
const { Kafka } = require('kafkajs');

// Create a Kafka client instance
const kafka = new Kafka({
  clientId: 'my-app',
  brokers: ['kafka1:9092', 'kafka2:9092']
});

// Create a producer instance
const producer = kafka.producer();

// Connect to Kafka
await producer.connect();

// Produce a message to a topic
await producer.send({
  topic: 'my-topic',
  messages: [
    { value: 'Hello Kafka!' }
  ]
});

// Disconnect from Kafka
await producer.disconnect();
```

Use the JavaScript Kafka Module FREE CODE!!

Identify Broker

Create Producer & Connect to Kafka

Send a Message to Kafka

Sample Kafka Messages

1.{"customer_id": "12345", "product_id": "67890", "action": "purchase", "timestamp": "2022-03-16T10:30:00Z"}

2.{"sensor_id": "9876", "measurement": 20.5, "unit": "Celsius", "timestamp": "2022-03-16T12:15:00Z"}

3.{"user_id": "abcd1234", "event_type": "click", "page_url": "https://example.com/home", "timestamp": "2022-03-16T14:20:00Z"}

4.{"order_id": "o123456", "customer_id": "c7890", "product_id": "p2345", "quantity": 2, "timestamp": "2022-03-16T16:45:00Z"}

5.{"sensor_id": "2345", "measurement": 80.2, "unit": "Fahrenheit", "timestamp": "2022-03-16T18:30:00Z"}

Eradani

Eradani

You Must…

- Produce Messages for Kafka from RPG/COBOL
- Format the Messages for Kafka
- Consume Messages from Kafka
- Translate Messages Coming From Kafka
- Handle the Event Driven to Procedural Transition
- Authenticate and Authorize
- Use the Kafka Communication Modules

Eradani

Setting up Kafka with RPG

- Need a Kafka Client
- Kafka Clients are very complex -> use open source!
- Major Kafka Clients sadly do not include RPG
- Connect RPG to Open Source and use open source Kafka clients!

Align with IBM's IBM i Development Strategy

The future of IBM i development is a blended environment

IBM i

World's Best RDBMS

COBOL+RPG

Lowest cost of ownership (TCO)

Reliability, securability, efficiency

Protection of investment

Open Source

Artificial Intelligence

Quantum Computing

Microservices / APIs

DevOps

Internet of Things

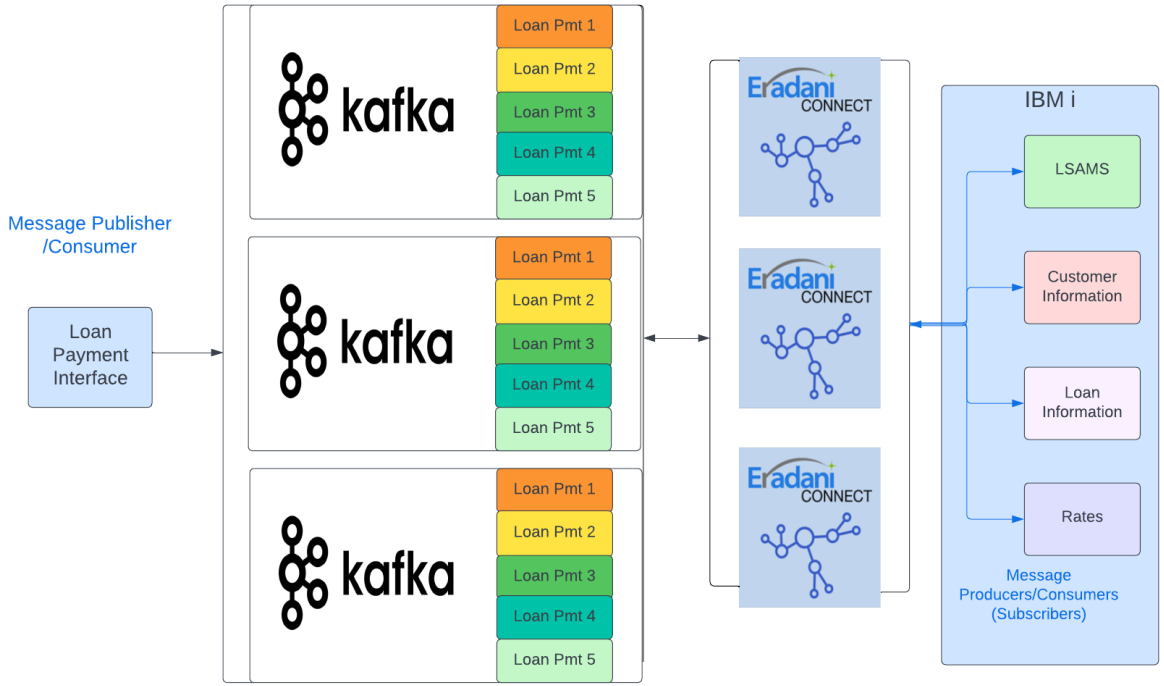
Web Technologies


Using the right tools for the job
will do more for your productivity
than hacking tools to do jobs
they were not designed to do.

Why JavaScript for Kafka?

- Use Kafka Modules
- Popular (easy to find resources)
- Easy to Learn
- Handle Kafka Objects
  - Very High Performance Transformations
- Designed for Event Driven, Asynchronous Processing

Managing Kafka Communications to IBM i



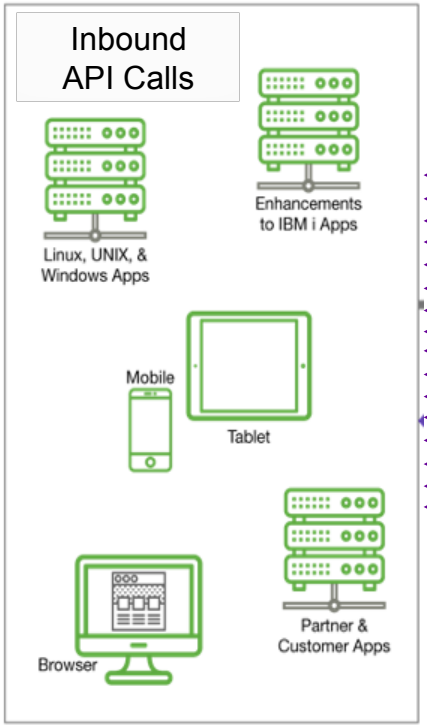
Eradani:

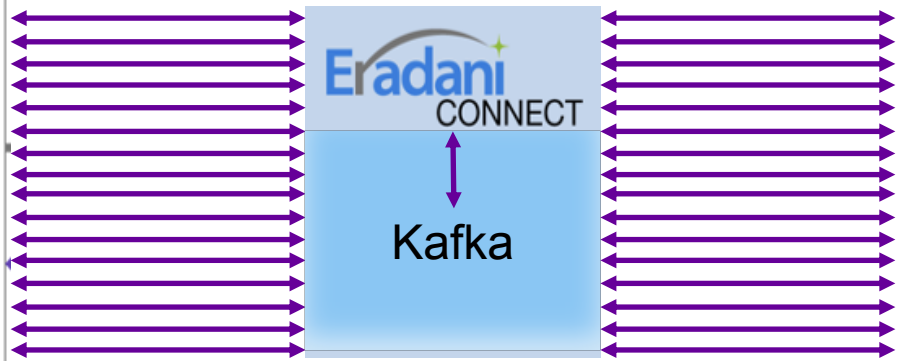
- Generate RPG Code and JavaScript Code to Send Kafka Messages
- Data Transformations
- IBM i Connections
- Kafka Message Formatting
- Monitoring
- Error Handling
- Event Driven to Procedural Translation
- Asynchronous Communication Management

High Performance to Match Kafka Performance!

Performance

With Kafka Link
8,000 Calls per Second





We can Add: More Servers, Optimization, Clustering & Load Balancing

Eradani

Indexing Basics

Speaker: Paul Tuohy

High Performance, Resilient APIs for Your IBM i Using Kafka

Speaker: Dan Magid

Paul's handout at
https://cutt.ly/IndexingBasics
or
https://cutt.ly/IndexingBasics_SiD

Sponsored by

Eradani

dan@eradani.com

See more Summit Lunch & Learn webinars at
SystemiDeveloper.com/LunchLearn

A Sample Table

Primary Key constraint provides an access path

Populated with a million rows

- Examples are only using the KEYID, SOMETEXT, SOMECODE and SOMEVALUE columns
- SOMETEXT values begin with ‘Low’, ‘Hit’, ‘Medium’ or ‘High’
  - 100 rows have a SOMETEXT value beginning with ‘Hit ’
- SOMECODE has only 4 values - “AA’, ‘BB’, ‘CC’ and ‘DD’

Stored procedure *fillindex*() used to populate table

- Code for stored procedure in notes

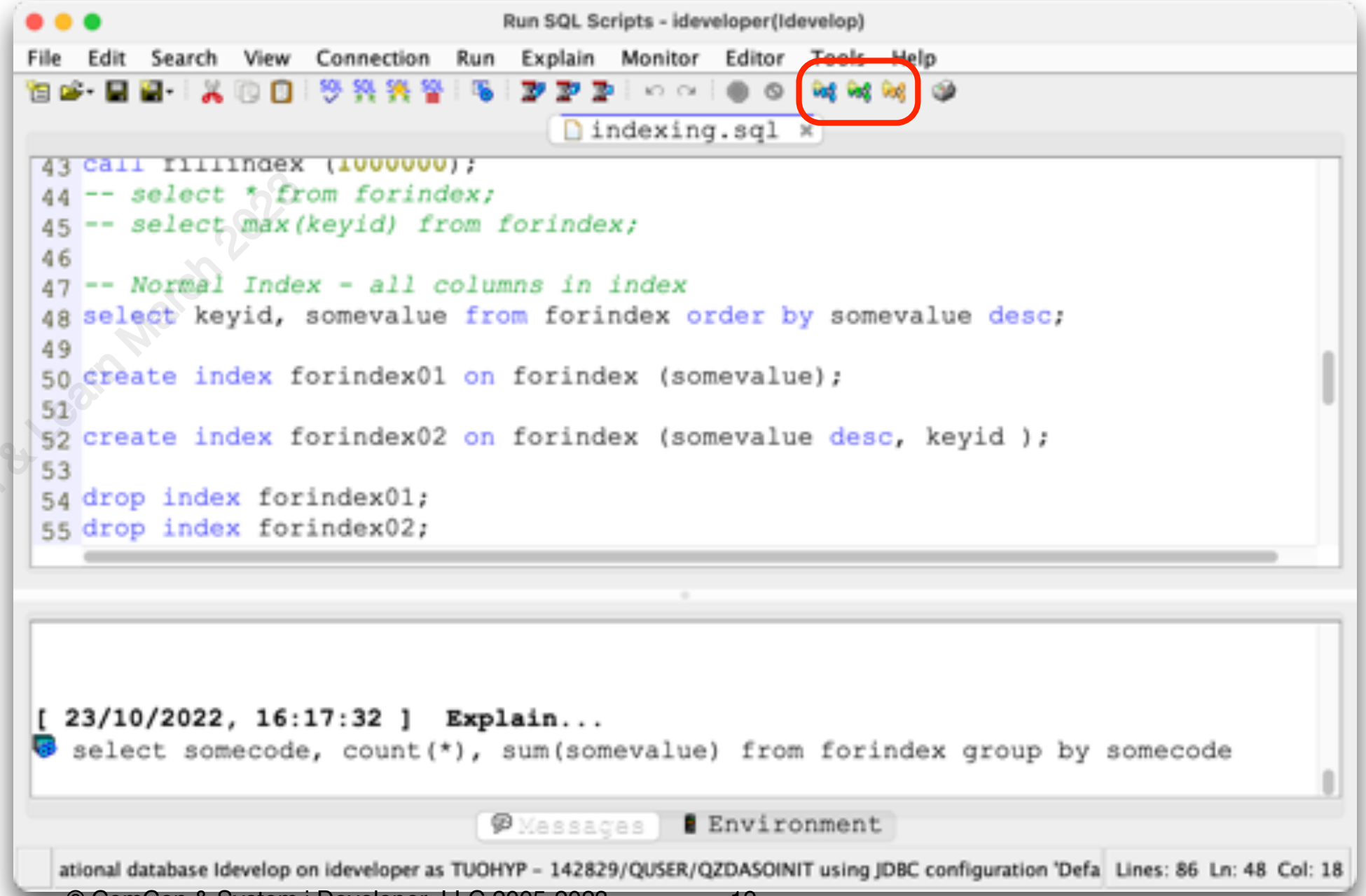

```
create or replace table forindex
(keyid integer generated always as identity,
 sometext varchar(20) allocate(20) not null default,
 somecode char(2) not null default,
 somevalue decimal(15, 0) not null default,
 other1 decimal(3, 0) not null default,
 other2 decimal(3, 0) not null default,
 other3 decimal(3, 0) not null default,
 constraint pk_forindex primary key (keyid)) ;
```

Stored procedure used to populate test table

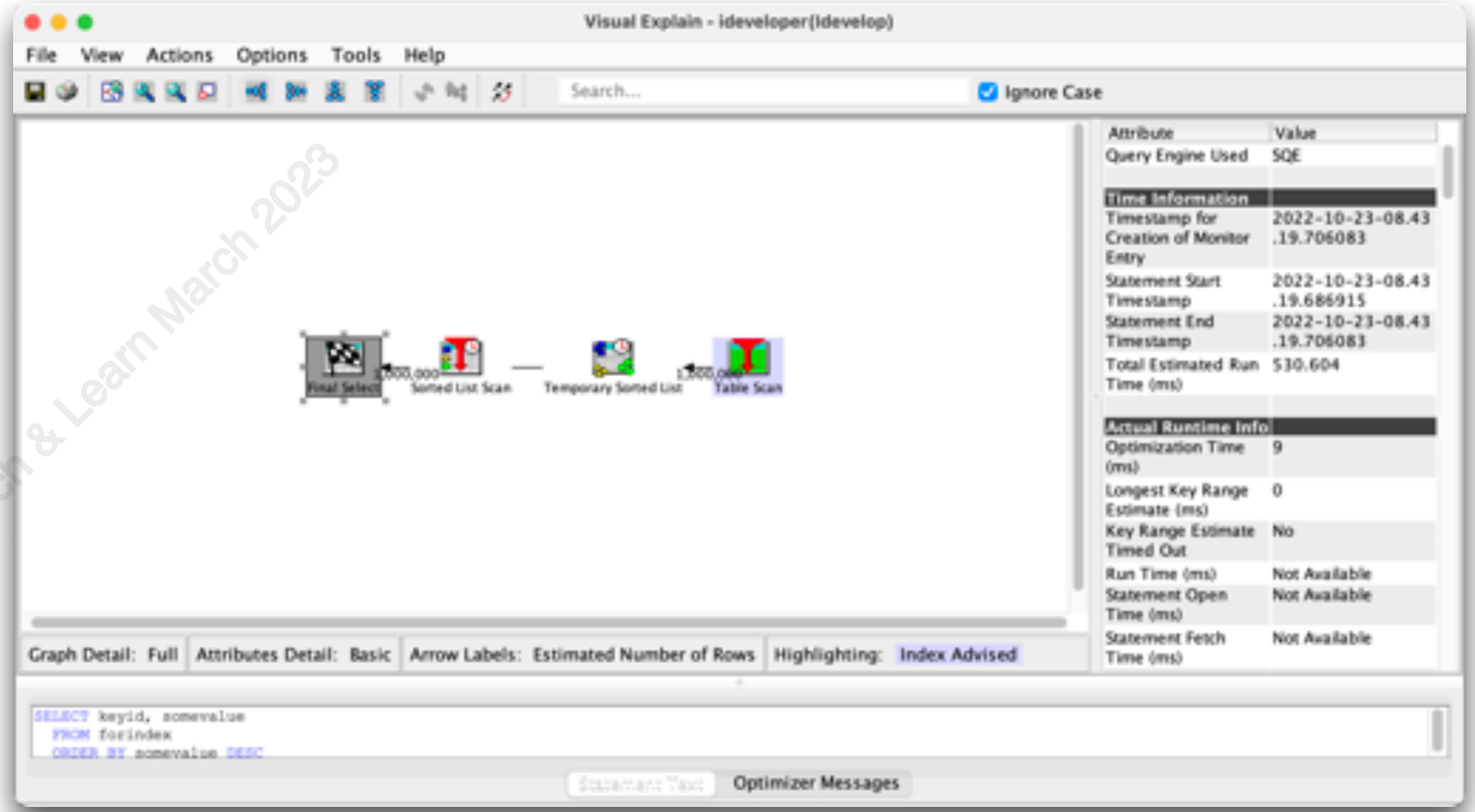
```
create or replace procedure fillindex
(in numrows integer)
begin
declare ii integer default 0;
declare sometext varchar(20) default '';
declare somecode char(2) default '';

while (ii < numrows) do
  set ii = ii + 1;
  case
    when ii < 10000 then
      set sometext = 'Low ' || char(ii);
      set somecode = 'AA';
    when ii < 10100 then
      set sometext = 'Hit ' || char(ii);
      set somecode = 'BB';
    when ii < 500000 then
      set sometext = 'Medium ' || char(ii);
      set somecode = 'CC';
    else
      set sometext = 'High ' || char(ii);
      set somecode = 'DD';
  end case;
  insert into forindex (sometext, somevalue, somecode)
    values (sometext, ii * 10, somecode);
end while;
end;
```

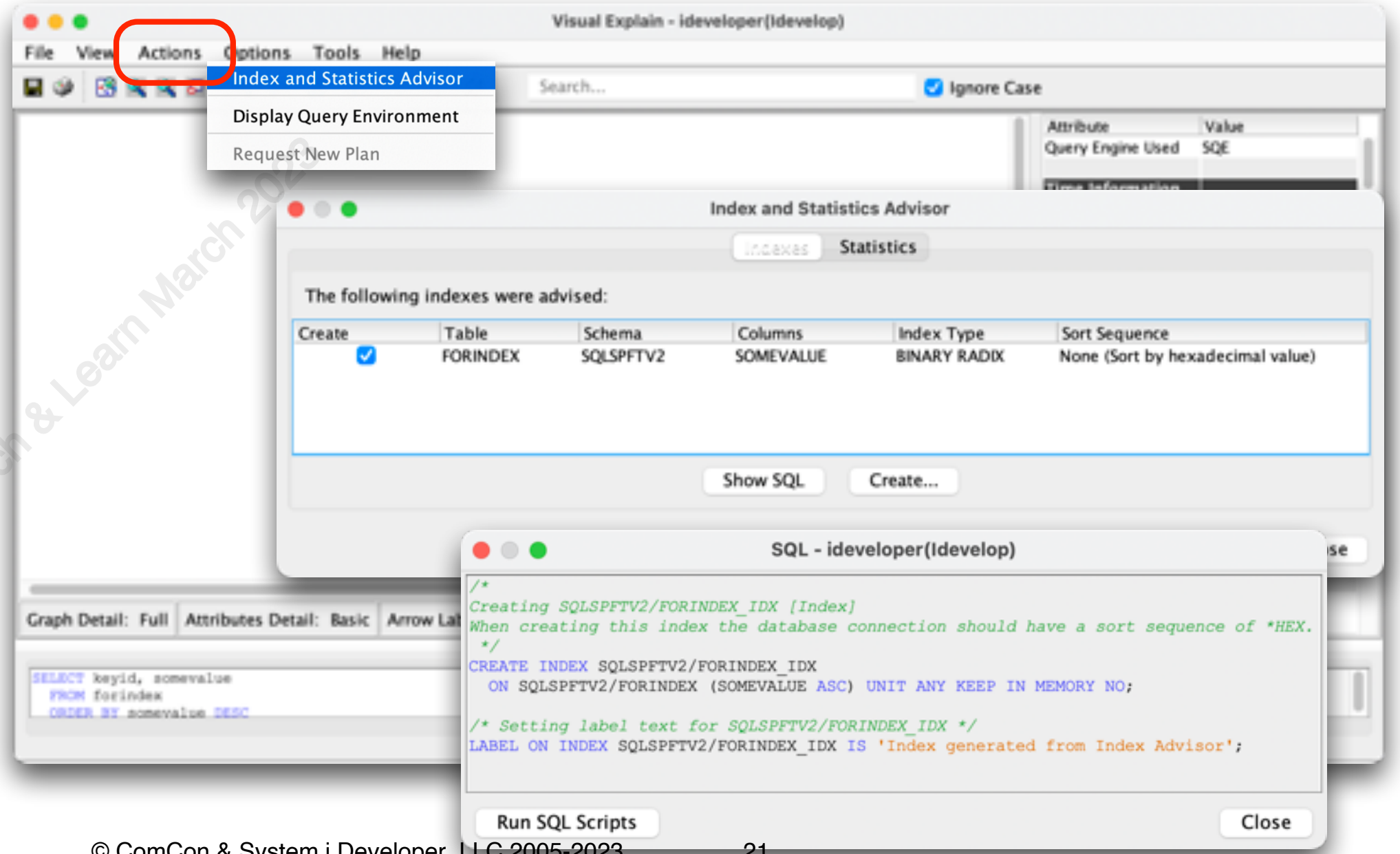
Visual Explain



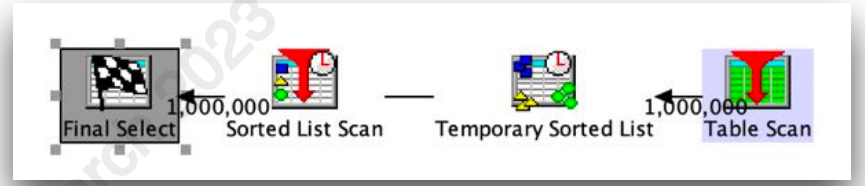
Visual Explain



Visual Explain and Advised Indexes



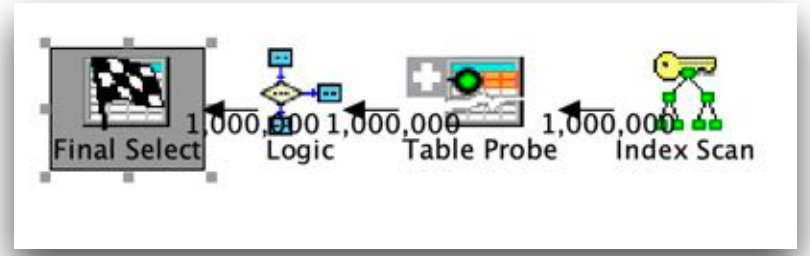
select keyid, somevalue from forindex order by somevalue desc;



Total Estimated Run Time (ms) 530.604

Index Advised

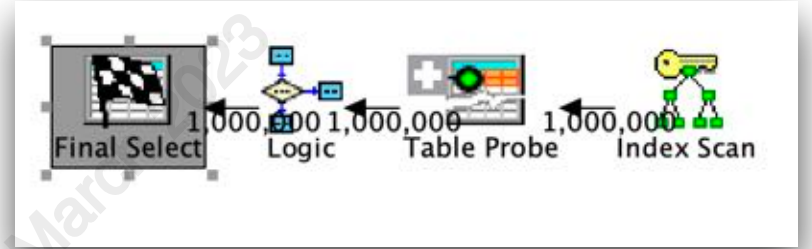
create index forindex01 on forindex (somevalue);



Total Estimated Run Time (ms) 1.711

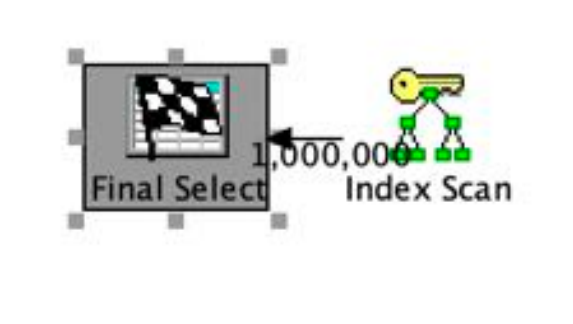
Basic Index - Beyond Advise

select keyid, somevalue from forindex order by somevalue desc;
create index forindex01 on forindex (somevalue);



Total Estimated Run Time (ms) 1.711

create index forindex02 on forindex (somevalue desc, keyid);

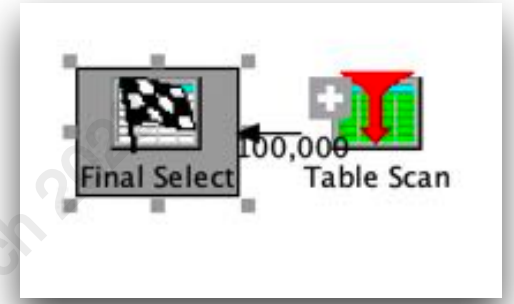


Total Estimated Run Time (ms) .43

Total Estimated Run Time (ms) 530.604

Derived Key Index

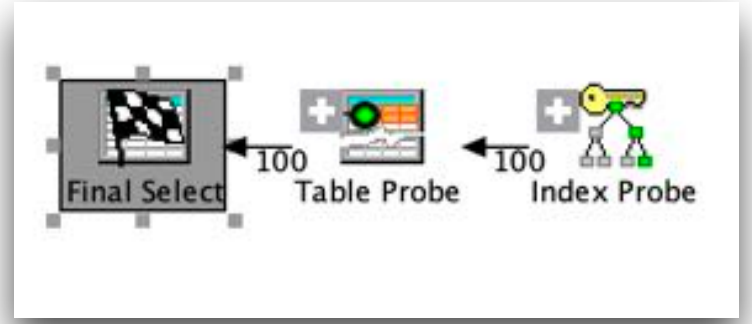
select * from forindex where lower(sometext) like 'hit%';



Total Estimated Run Time (ms) 83.966

No Index Advised

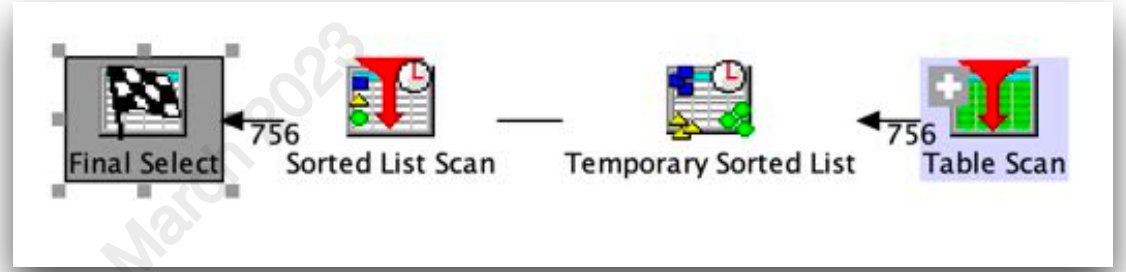
create index forindex03 on forindex (lower(sometext));



Total Estimated Run Time (ms) 1.287

Sparse Index

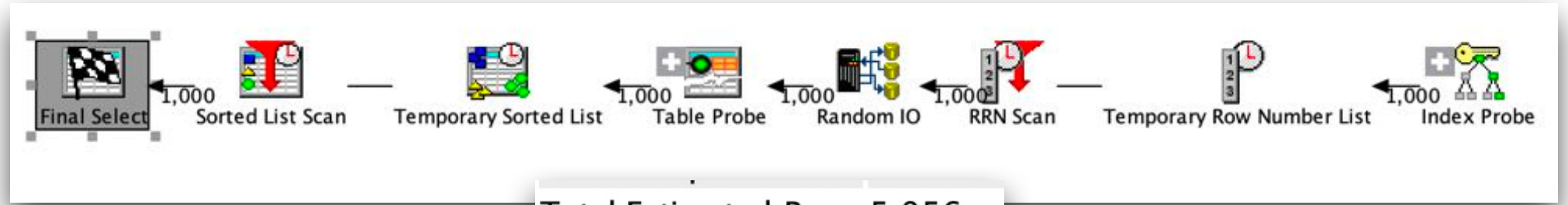
select keyid, sometext, somevalue from forindex
where somevalue <= 10000 order by keyid, somevalue;



Total Estimated Run Time (ms)	86.307

Indexes Advised

create index forindex04 on forindex (somevalue);
create index forindex05 on forindex (keyid, somevalue);

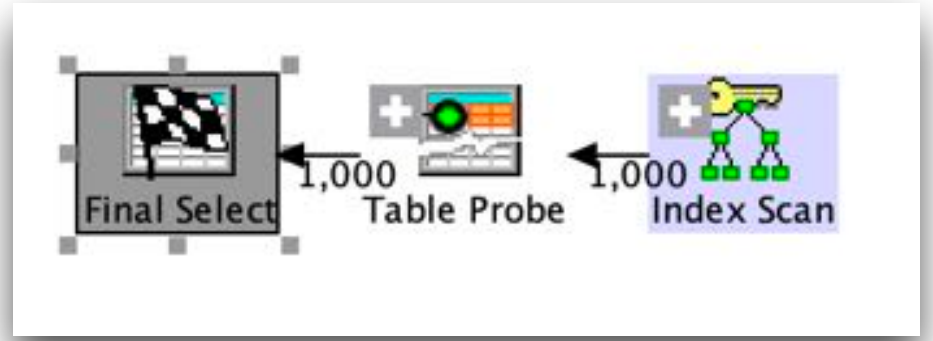


Total Estimated Run Time (ms)	5.956

Sparse Index

select keyid, sometext, somevalue from forindex
where somevalue <= 10000 order by keyid, somevalue;

drop index forindex04;
drop index forindex05;
create index forindex06 on forindex (keyid, somevalue)
where somevalue <= 10000;

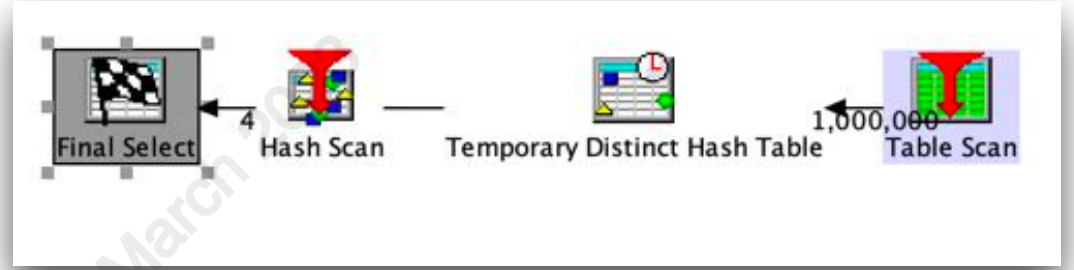


Total Estimated Run Time (ms)	1.291

Total Estimated Run Time (ms)	86.307

EVI With Aggregate Info

select somecode, count(*), sum(somevalue) from forindex group by somecode;



Total Estimated Run Time (ms) 394.32

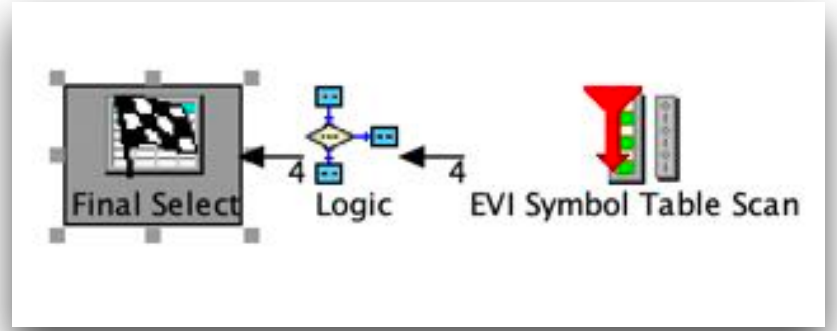
Index Advised

create index forindex07 on forindex (somecode);

Total Estimated Run Time (ms) 394.32

EVI With Aggregate Info

create encoded vector index forindex08 on forindex (somecode) include (count(*), sum(somevalue));



Total Estimated Run Time (ms) .426

Total Estimated Run Time (ms) 394.32

Last Bits

Avoid the following in WHERE and/or ORDER BY clauses

- numeric conversions
- arithmetic expressions
- character string padding
- the use of LIKE patterns beginning with % or _

References

- IBM Db2 for i indexing methods and strategies
  - https://www.ibm.com/support/pages/system/files/inline-files/Indexing%20and%20Statistics_1.pdf
- Creating an index strategy
  - https://www.ibm.com/docs/en/i/7.5?topic=optimization-creating-index-strategy
- Db2 for i Services
  - https://www.ibm.com/docs/en/i/7.5?topic=optimization-db2-i-services

Indexing Basics

Speaker: Paul Tuohy

High Performance, Resilient APIs for Your IBM i Using Kafka

Speaker: Dan Magid

Paul's handout at
https://cutt.ly/IndexingBasics
or
https://cutt.ly/IndexingBasics_SiD

Sponsored by

Eradani

dan@eradani.com

See more Summit Lunch & Learn webinars at
SystemiDeveloper.com/LunchLearn