

ORACLE®

# Do more with your Text Data in the Oracle Database

A primer for application developers

Roger Ford  
Principal Product Manager  
Oracle Server Technologies Division  
October 3rd, 2017

The logo for Oracle Open World, featuring the text "ORACLE OPEN WORLD" in red, stacked vertically, next to a red L-shaped graphic element.

ORACLE  
OPEN  
WORLD

October 1–5, 2017  
SAN FRANCISCO, CA

# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

**Note:** The speaker notes for this slide include instructions for when to use Safe Harbor Statement slides.

**Tip!** Remember to remove this text box.

# Program Agenda

- 1 Introduction to Oracle Text
- 2 App development: SQL or XML?
- 3 12.2 Review
- 4 18c Preview

# Program Agenda

- 1 Introduction to Oracle Text
- 2 App development: SQL or XML?
- 3 12.2 Review
- 4 18c Preview

# Text in your database?

- Most applications have some textual information
  - VARCHAR2, CLOB, even BLOB
- Very often, this text is not searchable, or only searchable by exact match
- Full-text search can make that text accessible and usable, enhancing the whole application
- Navigation by keywords and facets is often much quicker than navigation by hierarchy
- Oracle Text provides full text search right in the database



# Text Indexes

## The basics

- A text index indexes the **words** in a document set
- This is known as an "inverted index"
  - A table contains a set of documents, each of which is a list of words
  - The index is a set of words, each of which has a list of documents in which it occurs
- By looking up any word in the index we can find which documents it occurs in
- Oracle Text has a number of index types for different scenarios
  - Most common is CONTEXT indextype



# Creating an Oracle Text Index

```
CREATE INDEX prod_name_idx ON  
  product_information(product_name)  
  INDEXTYPE IS ctxsys.context ;
```

```
SELECT score(123), product_id, product_name  
  FROM product_information  
  WHERE contains (product_name,  
    'monitor NEAR full hd', 123)>0  
  ORDER BY score(99) DESC ;
```

SCORE(99)	PRODUCT_ID	PRODUCT_NAME
72	3331	Full HD Monitor 22 inch
56	3060	Monitor and TV combo, full HD

# Program Agenda

- 1 Introduction to Oracle Text
- 2 App development: SQL or XML?
- 3 12.2 Review
- 4 18c Preview

# Text Application Development

## Choosing your interface method

- The primary purpose for Oracle Text is full-text searching
- May be done through SQL or SQL and XML (Result Set Interface)
- Simple SQL - `SELECT ... WHERE CONTAINS(...)` - is most suited to:
  - Simple search/result scenarios
  - Extending existing SQL applications with full-text searches on VARCHAR/CLOB fields
- XML / Result Set Interface most suited to
  - Advanced search with metadata summaries and faceted navigation
  - Embedded advanced features such as sentiment analysis and collocations



# Result Set Interface

- Implemented by CTX\_QUERY.RESULT\_SET
  - call from SQL, JDBC, ODBC, etc.
- Requires an XML **Result Set Descriptor**
  - Describes what we want to fetch
- Returns an XML **Result Set**
  - Contains a hitlist and other info as requested

```
ctx_query.result_set('myIndex', '  
  <ctx_result_set_descriptor>  
    <hitlist start_hit_num="1" end_hit_num="1" >  
      <rowid />  
      <score />  
      <sdata name="title" />  
    </hitlist>  
    <count />  
  <ctx_result_set_descriptor>', RS);
```

RS:

```
<ctx_result_set>  
  <hitlist>  
    <hit>  
      <rowid>AAATWXAAGAAAQZ5AAM</rowid>  
      <score>23</score>  
      <sdata name="TITLE">My Word</sdata>  
    </hit>  
  
  </hitlist>  
  <count>130</count>  
</ctx_result_set>
```

# Program Agenda

- 1 Introduction to Oracle Text
- 2 App development: SQL or XML?
- 3 12.2 Review**
- 4 18c Preview

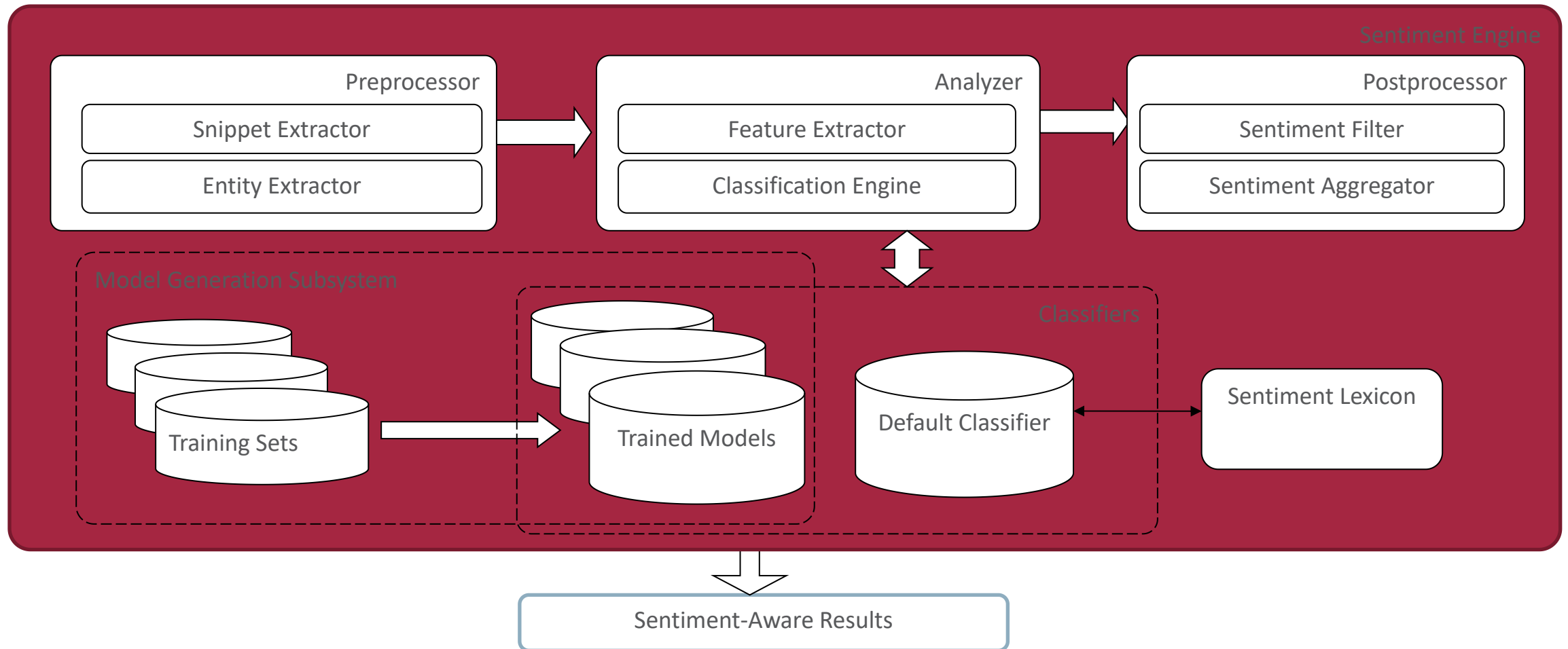
# Sentiment Analysis

Positive or negative sentiment with regard to search topic

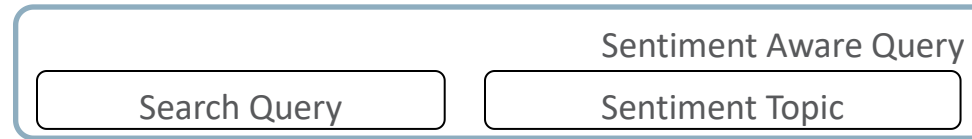


- Sentiment analysis (SA) traditionally only shows "good" or "bad" for whole document
- Oracle Text SA takes a topic or entity and performs SA for that
- e.g.
  - Is the plot good for this movie?
  - Are the hotel rooms clean / spacious / quiet?
- Has default "out of the box" capabilities
  - or can use classification techniques to train new models for higher accuracy and domain-specific applications

# Sentiment Architecture



# Sentiment Aware Queries



```
ctx_query.result_set('idx', `Camera AND Nikon  
S3`,  
<ctx_result_set_descriptor>  
  <hitlist order="SCORE DESC">  
    <sentiment classifier="camera">  
      <item topic="picture quality"/>  
      <item topic="lens" />  
    </sentiment>  
  </hitlist>  
  <group SA="picture quality">  
</ctx_result_set_descriptor>  
, :rs);
```

- Search Query
- Sentiment Topic
- Domain specific Classifier
- Sentiment Filter Criteria
- Sentiment Group Count



# Sentiment Analysis Notes

- Default classifier must use `AUTO_LEXER`.
  - Default sentiment analysis provided by Oracle Linguistic Technology
- Trained classifier may use *any* lexer
- Typical accuracy: Default: 73% Trained: 80%



# Collocates – the details

- **RSD:**

```
<ctx_result_set_descriptor>  
  <collocates radius="20" max_words="50" max_length="100"/>  
</ctx_result_set_descriptor>
```

- **RS:**

```
<ctx_result_set>  
  <collocates>  
    <collocation><word>AIR</word><score>75</score></collocation>  
    <collocation><word>MINI</word><score>7</score></collocation>  
    <collocation><word>64GB</word><score>7</score></collocation>  
    <collocation><word>PRO</word><score>7</score></collocation>  
  </collocates>  
</ctx_result_set>
```

# Program Agenda

- 1 Introduction to Oracle Text
- 2 App development: SQL or XML?
- 3 12.2 Review
- 4 18c Preview**

# Faceted Navigation

## Building Catalog-style applications with Oracle Text

- Oracle Text has traditional been used for document-centric applications
- Faceted navigation extends for catalog / webstore type applications
- Faceted navigation provides summary and drill-down capabilities
- 12.2 introduces new SDATA capabilities for faceted navigation in XML result set queries

**Price:**

- \$1 - \$5 (94)
- \$5 - \$10 (213)
- \$10 - \$50 (152)
- \$50 - \$100 (1)

**Item Type:**

- Music (265)
- Movie (192)
- HardGood (2)
- Game (1)

**Average Rating:**

- 4.5 - 5 stars (324)
- 3.5 - 4.4 stars (102)
- 2.5 - 3.4 stars (23)
- 1.5 - 2.4 stars (4)
- 0.5 - 1.5 stars (7)



**Disney's A Christmas Carol - Nintendo DS**  
By Sumo Digital Product SKU: 9417259  
**\$9.99**  
Average Review 3 Number of Reviews 2  
Disney's A **Christmas Carol** - Nintendo DS Take ... t



**Don't Open Till Christmas (DVD)**  
By (Unknown) Product SKU: 19812844  
**\$12.99**  
Average Review 5 Number of Reviews 1  
Don't Open Till **Christmas (DVD)** Movie Movie



**The Christmas Ornament (DVD)**  
By (Unknown) Product SKU: 25537374  
**\$4.99**  
Average Review 4 Number of Reviews 1  
The **Christmas Ornament (DVD)** Movie



**The Christmas Hope (DVD)**  
By (Unknown) Product SKU: 24263167  
**\$4.99**  
Average Review 4.8 Number of Reviews 12  
The **Christmas Hope (DVD)**



**Downton Abbey: Christmas at Downton Abbey**  
By (Unknown) Product SKU: 19870968  
**\$14.99**  
Average Review 4 Number of Reviews 3  
Downton Abbey: **Christmas at Downton Abbey (Blu**

# Setup for Faceted Navigation

- Add SDATA section group in index for each facet
  - use "optimized\_for" = "search"
- Add to Result Set Descriptor:

```
<group sdata = "color" topn="10">  
  <count/>  
</group>
```

- Returns:

```
<group sdata="color">  
  <group single="Red"> <count>123</count> </group>  
  <group single="Blue"> <count>26</count> </group>  
</group>
```

# Processing Facets

## Using returned facets in your application

### Use XMLTable:

```
select rs.face_label, rs.facet_count
from XMLTABLE(
  '/ctx_result_set/groups[@sdata="COLOR"]/group'
  PASSING xmltype(:rsout)
  COLUMNS
    facet_label  VARCHAR2(80) PATH '@single',
    facet_count  NUMBER PATH 'count/text()'
) AS rs
order by facet_count desc
```

- Or use XML capabilities of your application server

# 18c: New Substring Index

## Compact high performance index with K-Grams

- 12c Substring index uses rotated word-forms
  - oracle : <racle o> : <acle or> : <cle ora> : <le orac> : <e oracl>
  - Takes up lots of space, especially with long words
  - Garbage words major problem
  - AND separate structure for prefix indexing
- 18c uses K-grams
  - Handles prefix, suffix and double-truncated searches
  - Configurable length substrings : 2 – 5 characters
  - oracle : ^or : ora : rac : acl : cle : le\$
  - Each K-gram has its own (dense) postings list
  - No need to limit double-truncated searches



# 18c : Transactional Improvements

## OLTP text applications

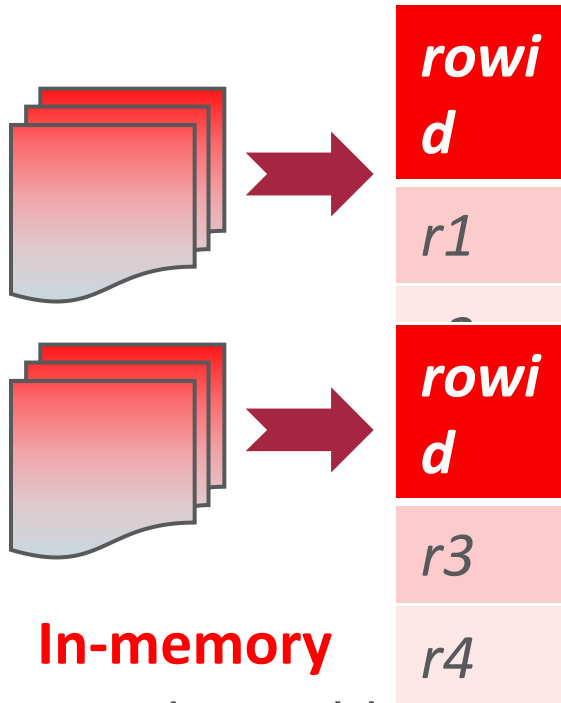
- Oracle Text traditionally targets document-centric applications
- Increasingly used for “small chunk” texts – Just Another Index
- In 18.1:
  - Remove functional and performance bottlenecks for small-chunk updates
  - Updating text-indexed columns should be as fast as b-tree indexed columns
  - No “\$R” table
  - Fully automatic indexes – no need for separate calls to SYNC\_INDEX or OPTIMIZE\_INDEX
    - SYNC(ON COMMIT) no longer costly
    - Automatic two-level index rebuilds postings on disk in optimal form
    - Automatic top-N token optimization keeps garbage manageable

# Scalable DML Architecture

## INSERT

## SYNC ON COMMIT

## AUTO MERGE



**In-memory**

Pending Table



token	token_info
night	..   ..   ..   ..
night	..   ..
day	..   ..   ..
night	101 <1,6>   102 <7>
day	101 <3,4>
wild	102 <4>
stormy	102 <6>

Staging table

New rows in Posting Lists



token	token_info
night	..   ..   ..   ..   101 <1,6>   102 <7>
day	..   ..   ..   101 <3,4>
wild	102 <4>
stormy	102 <6>

Defragmented Posting Lists

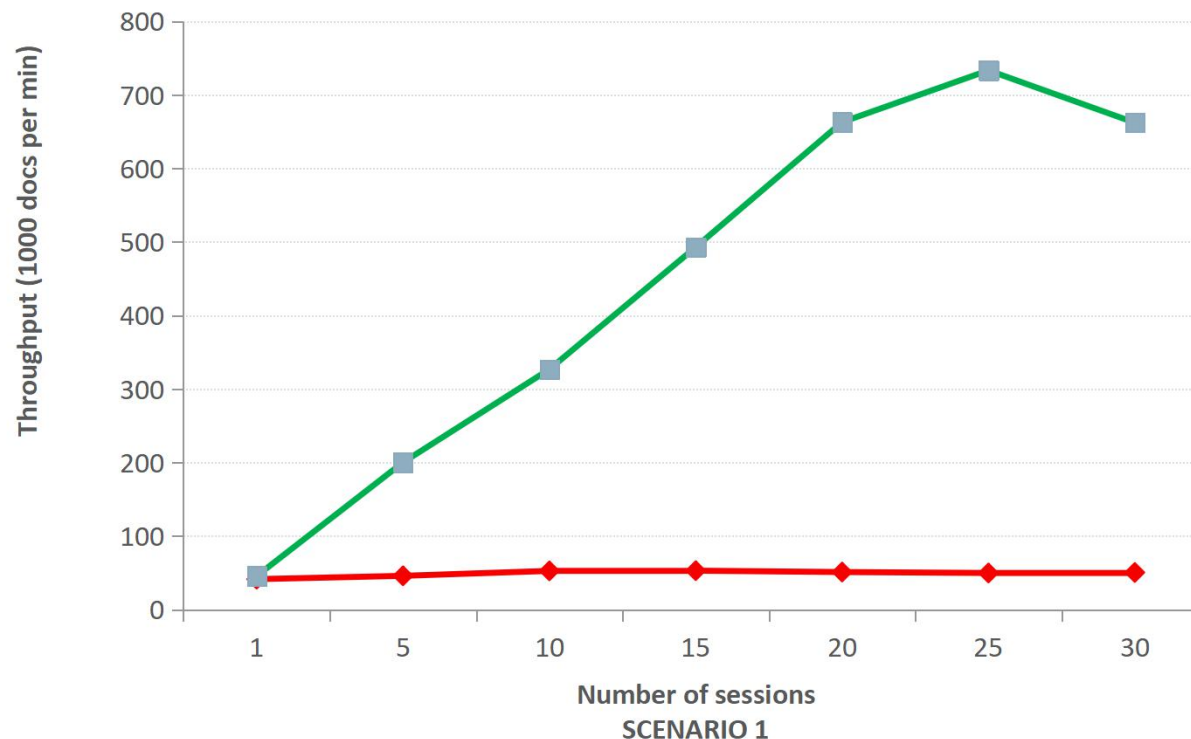
# Session scalability

## Removing bottlenecks

- No \$R table – major source of contention in old architecture
  - One row in \$R covers thousands of documents
- Local “pending” tables
  - No contention between different indexes waiting to store pending updates
- Gives major improvements in update scalability with multiple sessions

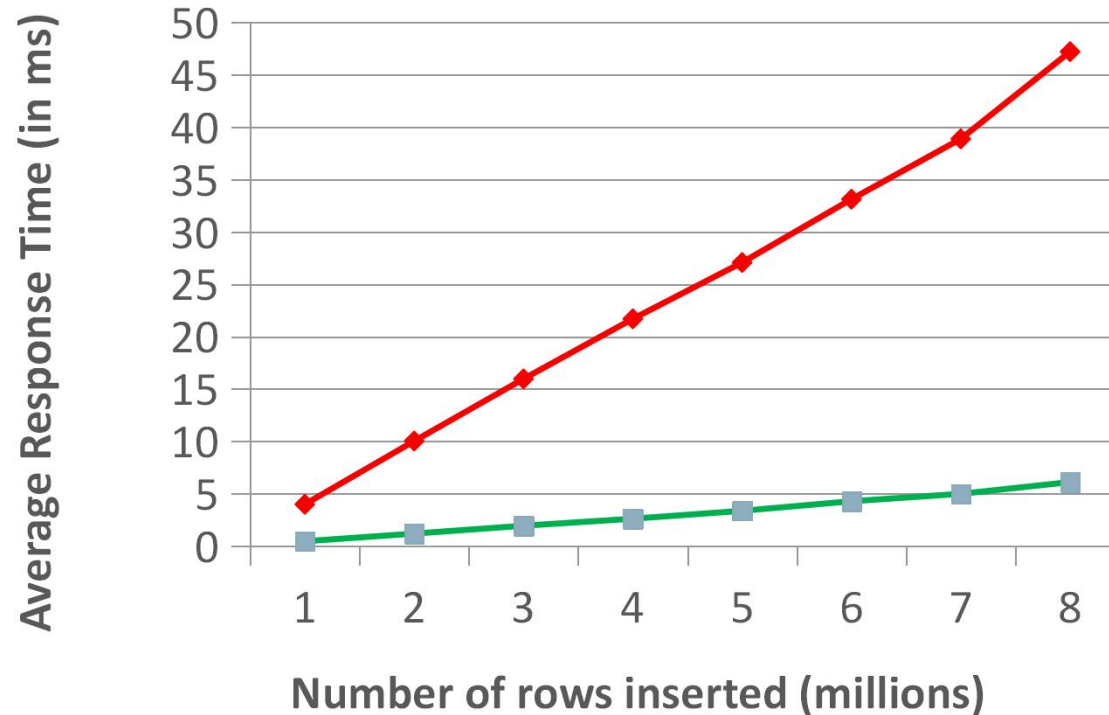
# Session scalability

- Inserts/updates by parallel sessions
- Green 18c, Red 12cR2
- 16-core machine
- Previously 50 TPS, regardless of sessions
- Now scales with sessions up to ~1.5 x core count



# Query Stability

- Without optimize, query performance drops rapidly as new data is added
- Auto-optimize in 18c vastly improves query stability
  - Auto-merge reduces fragmentation
  - Auto-top-token removes garbage from largest postings



# Demogrounds

- Come and find us:

**Moscone West, rear of hall**  
**Booth SOA 144 : Oracle Text, JSON and XML**

ORACLE®