

Declarative Search in APEX: Harnessing the Power of Search Configurations

11th March 2026
at

SOUG Day 2026 – Bern

in Bern, Switzerland

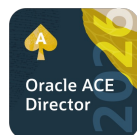
by

Niall Mc Phillips - Long Acre sàrl
niall.mcphillips@longacre.ch
@Niall_McP
@niallmcp.bsky.social



2

Niall McPhillips



SYMPOSIUM L2

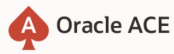


Long Acre sàrl

-  /  based in Geneva, Switzerland.
- Using Oracle DB since 1989
- Developing web applications with PL/SQL and Oracle DB since 1995
- Developing with APEX since 2004 (HTML DB 1.5)



6



The Oracle ACE Program

400+ technical experts helping peers globally



- The Oracle ACE Program recognizes and rewards community members for their technical and community contributions to the Oracle community
- 3 membership levels: Director, Pro, and Associate
- Nominate yourself or a colleague at ace.oracle.com/nominate
- Learn more at ace.oracle.com

✉ aceprogram_ww@oracle.com

Facebook.com/OracleACEs

X @oracleace

in Oracle ACE Program Group



7

SYMPOSIUM^{L2}

Created by the community, to support the community

Sharing of reliable knowledge
Supporting the various user groups and individuals



<https://sym42.org/>

8

Search Configuration - Standard



9

APEX Search Configurations (and more...) what we will see today

Standard search

Oracle Text (Basic, Search Engine and Expert)

Ubiquitous Search

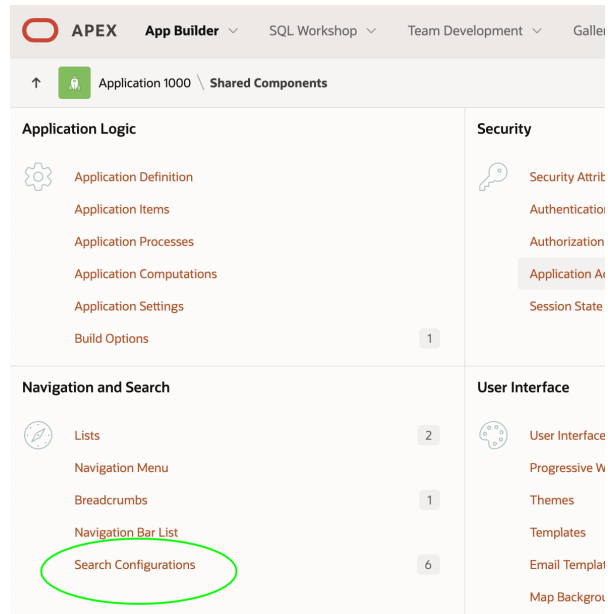
AI Vector Search

AI Hybrid Vector Search



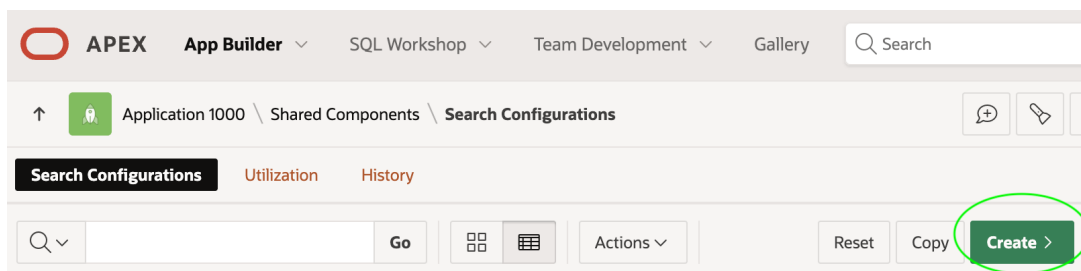
10

Shared Components - Search Configurations



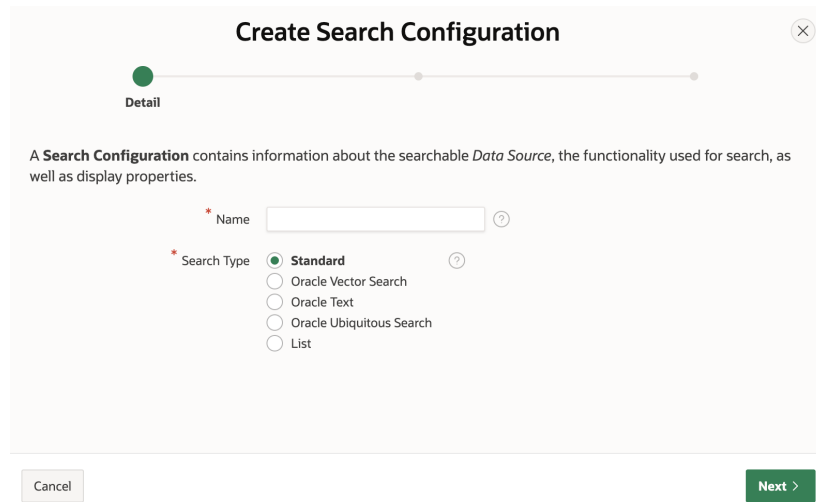
11

Create a Search Configuration



12

Creating a Search Configuration – Choose Search Type



Create Search Configuration

Detail

A **Search Configuration** contains information about the searchable *Data Source*, the functionality used for search, as well as display properties.

* Name

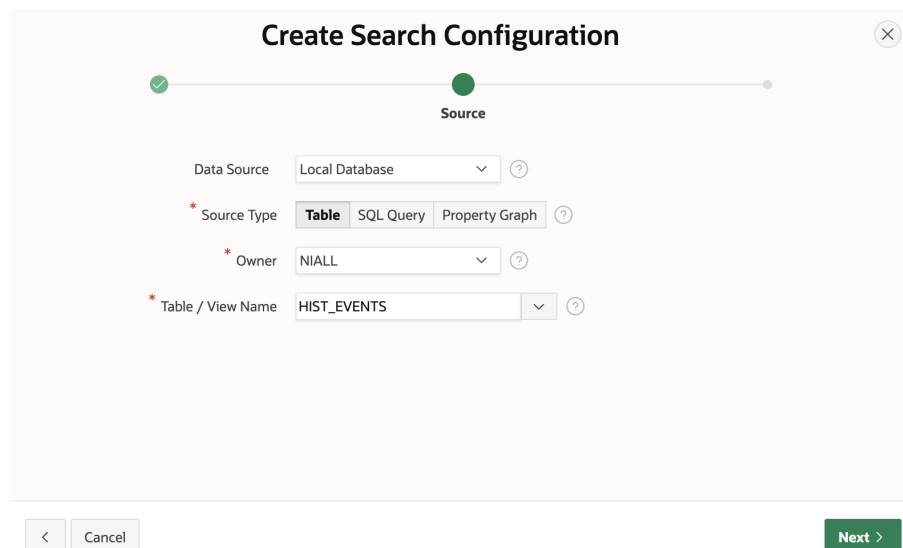
* Search Type

- ☒ Standard
- ☐ Oracle Vector Search
- ☐ Oracle Text
- ☐ Oracle Ubiquitous Search
- ☐ List

Cancel Next >

13

Creating a Standard Search Configuration – Data Source



Create Search Configuration

Source

Data Source

* Source Type **Table** SQL Query Property Graph

* Owner

* Table / View Name

< Cancel Next >

14

Creating a Standard Search Configuration – Key, Title and Description

Create Search Configuration

✓

✓

●

Column Mapping

* Primary Key Column

HIST_EVENT_ID (Number) ▼ ?

* Title Column

THEDATE (Varchar2) ▼ ?

Description Column

DESCRIPTION (Varchar2) ▼ ?

Icon Source

Initials

Icon Class

Icon Class Column

 ?

< Cancel

Create Search Configuration



15

Creating a Standard Search Configuration - Labels

Search Configuration: Standard Search 2

Cancel Delete Apply Changes

Show All

Settings

Subscription

Source

Column Mapping

Icon and Display

Link

Server-Side Condition

Advanced

Settings

Search Type

Standard ?

* Label

Standard Search 2 ?

Search Query Prefix

 ?

* Static ID

standard_search_2 ?



16

Creating a Standard Search Configuration – Searchable Columns

Source

Data Source: Local Database ?

* Source Type: **Table** SQL Query Function Body Returning SQL Property Graph ?

Table / View Owner: - Parsing Schema - ?

* Table / View Name: HIST_EVENTS ?

Where Clause: ?

Order By Clause: ?

Searchable Column(s):

HIST_EVENT_ID (Number)	THEDATE (Varchar2)
YEAR (Number)	DESCRIPTION (Varchar2)
SEARCH_VECTOR (Vector)	



17

Creating a Standard Search Configuration – Column Mapping

Column Mapping

* Primary Key Column ?	Primary Key Column 2 ?	
HIST_EVENT_ID (Number) v	- Select Column - v	
* Title Column ?	Subtitle Column ?	Description Column ?
THEDATE (Varchar2) v	- Select Column - v	DESCRIPTION (Varchar2) v
Custom Column 1 ?	Custom Column 2 ?	Custom Column 3 ?
- Select Column - v	- Select Column - v	- Select Column - v
Badge Column ?	Score Column ?	Last Modified Column ?
- Select Column - v	- Select Column - v	- Select Column - v



18

Creating a Standard Search Configuration - Link

Link

Link Type

Server-Side Condition

Condition Type

✓ - No Target -
Redirect to URL
Redirect to Page in this Application
Redirect to Page in a different Application



19

Creating a Search Page

Create a Page

Component Feature Legacy Pages

Form Interactive Grid Interactive Report Map Master Detail

Media List **Search Page** Smart Filters Timeline Tree

Unified Task List Wizard Workflow Console Plug-in Page

Cancel Help Create Page as Copy Next >



20

Creating a Search Page

Application 1000 \ Page 7: SS2

Search Page page created.

Page 7: SS2

- Pre-Rendering
- Components
 - Breadcrumb Bar
 - Breadcrumb
 - Search Field and Smart Filters
 - P7_SEARCH
 - Body
 - Search Results
 - Search Sources
 - Standard Search 2
- Post-Rendering

Layout Page Search Help

FULL WIDTH CONTENT

BODY

- Search Results
- SEARCH FIELD
- SEARCH RESULTS
- REGION CONTENT
- SUB REGIONS

FOOTER

DIALOGS, DRAWERS AND POPUPS

Regions Items Buttons

Region Attributes

Filter

Identification

Name Search Results

Title

Type Search

Source

Page Items to Submit

Layout

Sequence 20

Parent Region No Parent

Slot Body

Start New Row ☒

Row CSS Classes

Column Automatic

Column Span Automatic

Column CSS Classes

21

Creating a Search Page

Region Attributes

Filter

Settings

Search Page Item P7_SEARCH

Search as You Type ☒

Minimum Characters No minimum (0)

Appearance

Custom Layout ☐

Last Modified

Format Mask

Sort

Overall Sort - Select -

Pagination

Use Pagination ☒

Results per Page Type Static Value

Results per Page 15

Show Result Count No

22

Trying out the Search Page

Search

Danish

52 results

826 0826-06-24

The Danish king Harald Klak accepts Christianity.

838 0838

At Hingston Down, Egbert of Wessex beats the Danish and the West Welsh.

870 0870

The Great Summer Army invades England led by Bagsecg and conquers East Anglia the buildings destroyed by the Danish invaders include the abbey of Ely and the monastery of Peterborough.



23

Search Page – looking under the bonnet using debug mode

22	> APEX_APPLICATION_PAGE_REGIONS StandardSearchTable	3	
04	Searching on Standardsearch with query (Danish)	4	
50	search using query: Danish	4	
02	Generated Component SQL Query:	4	
76	<pre>select i.*, count(*) over () as APEX\$total_row_count from (select * from(select /** qb_name(apex\$inner) */* from(select x.* from "HIST_EVENTS" x)d)i)i where 1=1 and (((upper("THEDATE")like upper(:apex\$f1)) or (upper("DESCRIPTION")like upper(:apex\$f1)) or (upper("YEAR")like upper(:apex\$f1)))) order by YEAR</pre>	4	



24

Search Page - General



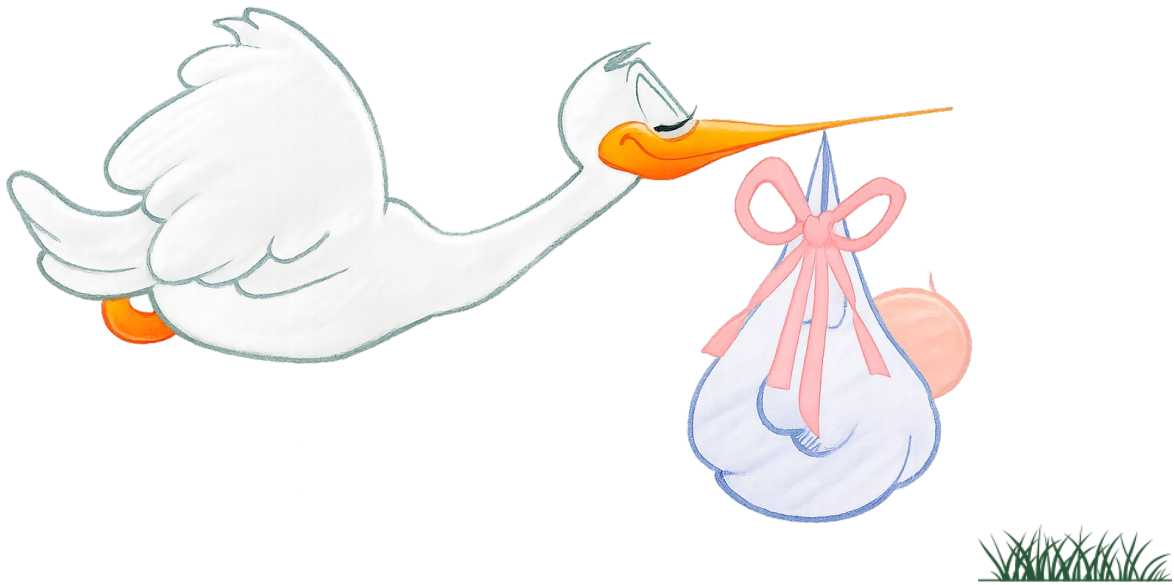
25

Search Configuration - Text Search - Basic



26

Where do Search Indexes come from?



27

Where do Search Indexes come from?

Oracle8 (1997) - Oracle ConText

Oracle8i (1999) - Oracle Intermedia Text.

Oracle9i (2001) - Oracle Text

Oracle23c (2023) - Search Indexes

28

Search Indexes - Built on a solid foundation



29

Using Search Indexes

- integral part of all Oracle database editions
- *really fast* and quite easy to start using
- just create an index and start searching
- index varchar2, XML, JSON, clobs and blobs (e.g., pdfs)
- uses the “contains” clause for querying
- allows AND/OR and more complex logic
- + many more advanced features...



30

Creating a simple Search Index

```
create search index indexname  
on tablename (columnname);
```



31

Creating a simple Search Index - examples

```
create search index si_judgments on judgments(description);
```

```
Index SI_JUDGMENTS created.
```

```
create search index si_hist_events on hist_events (description);
```

```
Index SI_HIST_EVENTS created.
```



32

Searching with contains

```
select * from tablename  
where  
contains(searchcolumn, 'searchtext') > 0;
```



33

Scoring search results

- The **score** of a search result gives an idea of the relevance of the result. High score indicates a higher relevance.
- Scores are always in the 1 to 100 range
- Scores have absolutely no meaning outside of their own query and cannot be compared between different queries, sub-queries or datasets.



34

Scoring search results - syntax

```
select score (1) , t.* from tablename t
where
contains (searchcolumn, 'searchtext', 1) > 0
order by 1 desc;
```

Note that the (1) in score(1) matches the ,1) in the contains clause



35

Oracle Text operator grammar and syntax Principal operators

- AND &
- OR |
- NOT ~
- ACCUM , at least one term matches
(it's like "OR" but
with higher scoring)



36

Some other operators

EQUIValence (=)

NEAR (;)

MINUS (-)

stem (\$)

Fuzzy

soundex (!)

and many more...

full details in Oracle Text Reference at:

<https://docs.oracle.com/en/database/oracle/oracle-database/23/ccref/index.html#Oracle%C2%AE-Text>



37

Creating a Text Search Configuration

Create Search Configuration

Detail

A **Search Configuration** contains information about the searchable *Data Source*, the functionality used for search, as well as display properties.

* Name

* Search Type ☐ Standard ☐ Oracle Vector Search ☒ **Oracle Text** ☐ Oracle Ubiquitous Search ☐ List

Cancel

Next >



38

Creating a Text Search Configuration

Create Search Configuration

✓

●

Source

Source Type **Table** SQL Query ?

* Table / View Owner NIALL ?

* Table / View Name HIST_EVENTS ?



39

Creating a Text Search Configuration

Create Search Configuration

✓

✓

●

Column Mapping

* Primary Key Column HIST_EVENT_ID (Number) ?

* Oracle Text Index Column DESCRIPTION (Varchar2) ?

* Title Column THEDATE (Varchar2) ?

Description Column DESCRIPTION (Varchar2) ?

Icon Source **Initials** Icon Class Icon Class Column ?

Cancel Create Search Configuration



40

Creating a Text Search Configuration – Basic

✓ Search configuration created. ✕

Search Configuration: Text Denmark

Cancel Delete Apply Changes

Show All Settings Subscription Source Column Mapping Icon and Display Link Server-Side Condition Advanced

Source

Source Type: **Table** SQL Query ⓘ

* Table / View Name: HIST_EVENTS ⓘ

Where Clause: ⓘ

Order By Clause: ⓘ

* Index Column Name: DESCRIPTION (Varchar2) ⓘ

Text Query Function:

- ✓ None
- Search Engine
- Expert Search
- Custom

 ⓘ

Column Mapping

* ⓘ



41

Trying out the Text Search Configuration

APEX search configs

Standard Oracle Text (Search Index) ▼

🔍 denmark and battle

13 results

- 1219-06-15**
Battle of Lyndanisse: Danish crusaders led by King Waldemar II conquer Tallinn. The Flag of Denmark allegedly falls from the sky during that battle.
- 1157-10-23**
Battle of Grathe Heath: A civil war in Denmark ends with the death of King Sweyn III. Valdemar I of Denmark becomes king of all Denmark and restores and rebuilds the ci
- 1471-10-10**
Battle of Brunkeberg in Stockholm: The forces of Regent of Sweden Sten Sture the Elder, with the help of farmers and miners, repel an attack by Christian I, King of Denr
- 1628-09-02**
Thirty Years' War: Wallenstein defeats Christian IV of Denmark's army in the Battle of Wolgast.
- 1807-09-02**
September 7 - Battle of Copenhagen: The Royal Navy bombards Copenhagen with fire bombs and phosphorus rockets to prevent Denmark from surrendering its fleet to Napoleon 30% of the city is destroyed and 2,000 citizens are killed.



42

Creating a Text Search Configuration

0.02036	0.00004	Generated Component SQL Query:	4	
0.02040	0.00233	<pre> select i.*, count(*) over () as APEX\$TOTAL_ROW_COUNT from (select * from(select /*+ qb_name(apex\$inner) */* from(select config_label,result_seq,primary_key_1,primary_key_2,title,subtitle,description,badge,last_modified,custom when custom_01 is not null or custom_02 is not null or custom_03 is not null then 'Y' end has_custom_attr from table(APEX_240200.wv_flow_search_api.search(p_search_static_ids => APEX_240200.wv_flow_t_varchar2(:apx\$sc1),p_search_expression =>:apx\$q,p_apply_order_bys =>:apx\$ordby,p_return_total_row_count => :apx\$cnt)))d)i)i where 1=1 order by "SCORE" desc nulls first </pre>	4	
0.02273	0.00016	Searching on Text 1 with query ({denmark} and {battle})	4	
0.02289	0.00021	Oracle Text Search query is "{denmark} and {battle}"	4	
0.02310	0.00002	Generated Component SQL Query:	4	
0.02312	0.01351	<pre> select i.*, count(*) over () as APEX\$TOTAL_ROW_COUNT from (select i.*, (SCORE(1)) "APEX\$ORATEXT_SCORE" from(select /*+ qb_name(apex\$inner) */* from(select x.* from "HIST_EVENTS" x where contains("DESCRIPTION",:apex\$f1,1)>0)d)i)i where 1=1 </pre>	4	
0.03663	0.00010	returned 13 rows.	4	



43

Search Configuration - Text Search – Search Engine



44

Creating a Text Search Configuration – Search Engine

Firstly, let's talk about "Query Relaxation":

enables your application to execute the most restrictive version of a query first and progressively relax the query until the required number of hits are obtained.

Example: searching for **green** and **pen**

```
select id from docs where CONTAINS (text,
'<query>
  <textquery lang="ENGLISH" grammar="CONTEXT">
    <progression>
      <seq>{green} {pen}</seq>
      <seq>{green} NEAR {pen}</seq>
      <seq>{green} AND {pen}</seq>
      <seq>{green} ACCUM {pen}</seq>
    </progression>
  </textquery>
  <score datatype="INTEGER" algorithm="COUNT"/>
</query>')>0;
```



45

Creating a Text Search Configuration – Search Engine

Source Type: **Table** SQL Query ?

* Table / View Name: HIST_EVENTS ?

Where Clause: ?

Order By Clause: ?

* Index Column Name: DESCRIPTION (Varchar2) ?

Text Query Function: ☒ Search Engine ?

Expert Search

Custom



46

Trying out the Text Search Configuration – Search Engine

APEX search configs

Standard

Oracle Text (Search Index) ▾

Search Engine

1825-02-03

Vendsyssel-Thy, once part of the Jutland peninsula that formed westernmost Denmark, becomes an island after a flood drowns its 1ampnbspm wide isthmus.

1849-07-06

The Danish Army beats the Prussian army at Fredericia, Jutland, thereby putting an end to the Prussian/Danish War until 1864.

1916-05-31

June 1 - Battle of Jutland: The Royal Navy and the German Navy battle to a draw.

1948-09-06

Juliana becomes Queen of the Netherlands.



47

Creating a Text Search Configuration

Oracle Text Search query is "jutland"	4	☐
Generated Component SQL Query:	4	☐
<pre>select i.* from (select i.*,(SCORE(1)) "APEX\$ORATEXT_SCORE" from(select /** qb_name(apex\$inner) */* from(select x.* from "HIST_EVENTS" x where contains("DESCRIPTION",APEX_240200.www_flow_search_api.query_search_engine(:apex\$f1),1)>0)d)i)i where 1=1</pre>	4	☐
Generated Oracle Text query length: 143	4	☐
Generated Oracle Text query: "<query><textquery><progression><seq>{jutland}</seq><seq>\${jutland}</seq><seq>FUZZY({jutland},40,1000,W)</seq></progression></textquery></query>"	4	☒



48

Search Configuration - Text Search – Expert



49

Creating a Text Search Configuration – Expert

This function supports more advanced search operators for precise queries. It allows exact matching, Boolean operators (AND, OR, NOT), precedence with parentheses and proximity search.

Source

Source Type **Table** SQL Query ?

* Table / View Name HIST_EVENTS ?

Where Clause

Order By Clause

* Index Column Name

- None ?
- Search Engine ?
- ✓ Expert Search ?

Text Query Function



50

Creating a Text Search Configuration – Expert

APEX search configs

Standard Oracle Text (Search Index) ▼

Q {denmark & battle} or {beer}

- 11 1268-10-29**
In France, the use of hops as the exclusive flavoring agent used in the manufacture of beer is made compulsory.
- 11 1157-10-23**
Battle of Grathe Heath: A civil war in Denmark ends with the death of King Sweyn III. Valdemar I of Denmark becomes king of all Denmark and restores and rebuilds
- 11 1383-10-22**
Löwenbräu beer is first brewed.
- 11 1814-10-17**
London Beer Flood: A large vat full of porter (beer) owned by Meux's Brewery of London bursts, demolishing buildings and killing nine.
- 11 1939-11-08**
WWII: In Munich, an attempt to kill Adolf Hitler is made by Georg Elser while Hitler is celebrating the 16th anniversary of the Beer Hall Putsch.
- 10 1265-08-04**
The brewing of Budweiser Budvar beer begins in Bohemia (now part of the Czech Republic) Budweiser Budvar has been produced continuously there to this day.
- 10 1811-03-27**
Battle of Anholt: The British Navy defeats Denmark.



51

Search Configuration - Ubiquitous Search



52

26ai - Ubiquitous Search



53

26ai Ubiquitous Search

With 26ai Ubiquitous Search we can search across multiple, unrelated data sources



54

26ai Ubiquitous Search

DBMS_SEARCH



55

26ai Ubiquitous Search – Creating an index

First, we'll create the index:

dbms_search.create_index('UB1SEARCH');



56

26ai Ubiquitous Search – adding data sources

Next, let's add two completely unrelated data sources:

```
dbms_search.add_source('UB1SEARCH', 'ANIMAL_FACTS');
```

```
dbms_search.add_source('UB1SEARCH', 'HIST_EVENTS');
```



57

26ai Ubiquitous Search – creating a search config

The image shows a 'Create Search Configuration' dialog box. At the top, there is a progress bar with three dots; the first dot is green and labeled 'Detail'. Below the progress bar, a text box explains: 'A Search Configuration contains information about the searchable Data Source, the functionality used for search, as well as display properties.' The form contains two fields: 'Name' with a text input and a help icon, and 'Search Type' with a list of radio buttons. The radio buttons are: 'Standard', 'Oracle Vector Search', 'Oracle Text', 'Oracle Ubiquitous Search' (which is selected and has a green highlight), and 'List'. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Next >'.



58

26ai Ubiquitous Search – creating a search config

Create Search Configuration ✕

✓ ——— ● Source ———

* Search Index Name ⓘ

* Search Source Name ⓘ

< Cancel **Next >**

59

26ai Ubiquitous Search – creating a search config

Create Search Configuration ✕

✓ ——— ● Source ———

* Search Index Name ⓘ

* Search Source Name ⓘ

< Cancel **Next >**

60

26ai Ubiquitous Search – creating a search config

Create Search Configuration

Progress: 100% (3 steps)

Column Mapping

* Primary Key Column: ANIMAL_FACT_ID (Number) ?

* Title Column: ANIMAL_NAME (Varchar2) ?

Description Column: TEXT (Varchar2) ?

Icon Source: **Initials** Icon Class: Icon Class Column ?

< Cancel Create Search Configuration



61

26ai Ubiquitous Search – creating a search config

Subscribe From: [dropdown] ?

" copy of this Search Configuration.

itions subscribe to this configuration.

* Search Index Name: UBISEARCH ?

* Search Source Name: ANIMAL_FACTS ?

Order By Clause: [text area] ?

Text Query Function: **None** ?
Search Engine
Expert Search
Custom

* Primary Key Column: ANIMAL_FACT_ID (Number) ?

Primary Key Column 2: - Select Column - ?

* Title Column: ANIMAL_NAME (Varchar2) ?

Subtitle Column: - Select Column - ?

Description Column: TEXT (Varchar2)



62

26ai Ubiquitous Search – creating a search config

Ubiquitous Search with APEX Search Configurations

Quick Demo



63

26ai Ubiquitous Search – extending the search

*Extending the search to multiple tables
(does not use Search Configurations)*



64

26ai Ubiquitous Search – describe the table created

```
SQL> desc ublsearch
```

Name	Null?	Type

METADATA	NOT NULL	JSON
DATA		JSON
OWNER		VARCHAR2 (128)
SOURCE		VARCHAR2 (128)
KEY		VARCHAR2 (1024)



65

26ai Ubiquitous Search – explore the table created

Let's take a look at the metadata

```
select metadata from ublsearch  
where contains(data, 'brain') > 0;
```



66

26ai Ubiquitous Search – table created

Let's take a look at the metadata

```
{ "OWNER": "NIALL", "SOURCE": "HIST_EVENTS", "KEY": { "HIST_EVENT_ID": 4169 } }  
{ "OWNER": "NIALL", "SOURCE": "HIST_EVENTS", "KEY": { "HIST_EVENT_ID": 34276 } }  
{ "OWNER": "NIALL", "SOURCE": "HIST_EVENTS", "KEY": { "HIST_EVENT_ID": 24285 } }  
{ "OWNER": "NIALL", "SOURCE": "HIST_EVENTS", "KEY": { "HIST_EVENT_ID": 24487 } }  
....  
{ "OWNER": "NIALL", "SOURCE": "ANIMAL_FACTS", "KEY": { "ANIMAL_FACT_ID": 4193 } }  
{ "OWNER": "NIALL", "SOURCE": "ANIMAL_FACTS", "KEY": { "ANIMAL_FACT_ID": 4228 } }  
{ "OWNER": "NIALL", "SOURCE": "ANIMAL_FACTS", "KEY": { "ANIMAL_FACT_ID": 4230 } }
```



67

26ai Ubiquitous Search – getting to the data

We can extract the keys using JSON_TABLE

```
...  
  from ublsearch us  
  cross join json_table(us.metadata, '$.KEY[*]'  
    columns (animal_fact_id number path '$.ANIMAL_FACT_ID',  
             hist_event_id  number path '$.HIST_EVENT_ID')) j  
...
```



68

26ai Ubiquitous Search – getting to the data

Joining with the underlying tables...

```
...  
  from ublsearch us  
  cross join json_table(us.metadata, '$.KEY[*]'  
                        columns (animal_fact_id number path '$.ANIMAL_FACT_ID',  
                                hist_event_id number path '$.HIST_EVENT_ID')) j  
  left outer join hist_events h on (h.id = j.hist_event_id)  
  left outer join animal_facts af on (af.animal_fact_id =  
    j.animal_fact_id)  
...
```



69

26ai Ubiquitous Search – displaying the data

Join with the underlying tables...

```
...  
select case  
  when j.animal_fact_id is not null then 'Animal Fact'  
  else 'Historical Event'  
  
end as result_type,  
coalesce(j.animal_fact_id, j.hist_event_id) as theId,  
coalesce(h.description, af.text) as theText  
from ublsearch us  
cross join ... (see in previous slide)
```



70

26ai Ubiquitous Search – displaying the data

Creating a view for other developers

```
create or replace view vw_ubsearch as
select case
    when j.animal_fact_id is not null then 'Animal Fact'
    else 'Historical Event'
end as result_type,
coalesce(j.animal_fact_id, j.hist_event_id) as theId,
coalesce(h.description, af.text) as theText,
us.data
from ublsearch us
cross join json_table(us.metadata, '$.KEY[*]'
    columns (animal_fact_id number path '$.ANIMAL_FACT_ID',
             hist_event_id number path '$.HIST_EVENT_ID')) j
left outer join hist_events h on (h.hist_event_id = j.hist_event_id)
left outer join animal_facts af on (af.animal_fact_id = j.animal_fact_id);
```



71

26ai Ubiquitous Search – multi-table

Multi-Table Ubiquitous Search with APEX

Quick Demo



72

Search Configuration - AI Vector Search



73

What is AI Vector Search

AI Vector Search enables you to search structured and unstructured data based on its semantics or meaning, in addition to its values.



74

Vectors

Vectors are derived from the ***semantic content*** of your data, rather than the underlying words or image parts.

For example, a search for the term ***architecture*** could also find results such as *buildings*, *construction* or even *a famous architect*.

In this respect they are quite different to other search options such as Oracle Search Indexes.



75

Vectors

From a simplistic point of view, a vector is simply a list of numbers.

The number of dimensions that a vector has is how many numbers it contains.

[0.8763, 1.8476, 0.2977, 2.8686,]



76

Vector Similarity

Vector similarity is calculated using the mathematical distance between two vectors

$[0.8763, 1.8476, 0.2977, 2.8686, \dots]$
 $[0.7836, 1.6748, 0.7729, 2.6868, \dots]$



77

Where do I get my vectors – Vector Embeddings

The creation / calculation of a vector from data is called ***EMBEDDING***

There are many different embedding models, including those provided by OpenAI, Cohere and many open-source models.

The model you choose will be influenced by the data that you are vectorising: text, images, video, sound,



78

Where do we get our vectors – Vector Embeddings

For the purposes of today's presentation, we'll be using a model downloaded from Hugging Face called:

all_MiniLM_L12_v2.onnx



79

AI Vector Search Importing pre-trained models

Import pre-trained embedding ML models for vector generation within the DB

- Open Neural Network Exchange (ONNX) is an open format built to represent machine learning models
- <https://onnx.ai/>



80

Models in the DB

- import an embedding model into the database
- ONNX models can be found on Hugging Face and other platforms
- To import a model use:
dbms_vector.load_onnx_model_cloud
- The model used here: *all_MiniLM_L12_v2.onnx*

81

Models in the DB

- Viewing models loaded in the DB

```
select *  
  from user_mining_models  
 where model_name='ALL_MINILM_L12_V2';
```

MODEL_NAME	MINING_FUNCTION	ALGORITHM	ALGORITHM_TYPE	CREATION_DATE	BUILD_DURATION	MODEL_SIZE
ALL_MINILM_L12_V2	EMBEDDING	ONNX	NATIVE	20-MAY-25	0	133322334

82

Let's look at the new data type: VECTOR

Since 23ai VECTOR is a new data type fully integrated into the database.

```
create table my_tab
(my_id integer,
 my_text varchar2(500),
 my_vector vector);
```

83

Embeddings – table of sentences

We will use a table called SENTENCES which contains 150 ChatGPT-generated sentences. 100 about cities in Europe and 50 about animals.

```
SQL> desc sentences
Name          Null?     Type
-----
ID             NUMBER
THE_TEXT      NOT NULL VARCHAR2(255)
THE_VECTOR    VECTOR(*,*,*)
```

ID	THE_TEXT	THE_VECTOR
1	Ljubljana, Slovenia is admired for its charming city center and lively atmosphere.	(null)
2	Split, Croatia is famous for the ancient Diocletian's Palace and its Adriatic coast.	(null)
3	Ljubljana, Slovenia is recognized for its friendly people and green spaces.	(null)
4	Sofia, Bulgaria is celebrated for its Orthodox churches and vibrant city life.	(null)
5	Bucharest, Romania is known for its mix of historical and modern architecture.	(null)
6	Belgrade, Serbia is admired for its fortresses and riverside setting.	(null)
7	Sarajevo, Bosnia and Herzegovina is famous for its diverse culture and historical significance.	(null)
8	Tirana, Albania is recognized for its colorful architecture and dynamic cultural scene.	(null)
9	Skopje, North Macedonia is noted for its eclectic blend of old and new buildings.	(null)
10	Pristina, Kosovo is admired for its youthful energy and growing arts scene.	(null)

85

Embeddings - updating the embedding vector

update all rows in the sentences table to populate their vectors

update sentences s

set the_vector

= vector_embedding

(all_minilm_l12_v2

using s.the_text as data);



86

We can now use the embedding

View our sentences table with its vectors populated

```
sql> select id, the_text, the_vector from sentences fetch first 10 rows only;
```

ID	THE_TEXT	THE_VECTOR
1	Ljubljana, Slovenia is admired for its charming city center and lively atmosphere.	[3.92498486E-002,3.8745027E-002,-1.1
2	Split, Croatia is famous for the ancient Diocletian's Palace and its Adriatic coast.	[3.04311961E-002,-4.54256348E-002,-
3	Ljubljana, Slovenia is recognized for its friendly people and green spaces.	[3.85775864E-002,4.74944785E-002,-1
4	Sofia, Bulgaria is celebrated for its Orthodox churches and vibrant city life.	[1.3237983E-002,-7.41957454E-003,-1
5	Bucharest, Romania is known for its mix of historical and modern architecture.	[7.14117195E-003,-1.28046563E-003,-
6	Belgrade, Serbia is admired for its fortresses and riverside setting.	[5.60169034E-002,7.53022218E-003,-2
7	Sarajevo, Bosnia and Herzegovina is famous for its diverse culture and historical significance.	[5.80895729E-002,-1.25712929E-002,-
8	Tirana, Albania is recognized for its colorful architecture and dynamic cultural scene.	[2.69968752E-002,-2.40882835E-003,-
9	Skopje, North Macedonia is noted for its eclectic blend of old and new buildings.	[3.20351236E-002,1.77446958E-002,-2
10	Pristina, Kosovo is admired for its youthful energy and growing arts scene.	[4.37207781E-002,1.52237155E-002,-2

10 rows selected.



87

New Vector function – VECTOR_DISTANCE

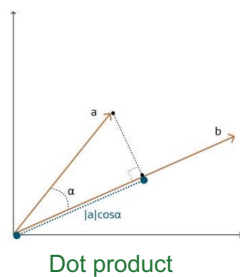
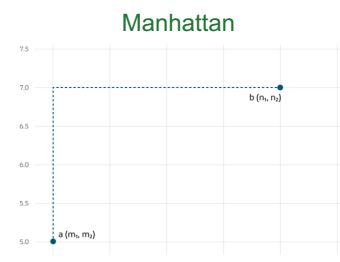
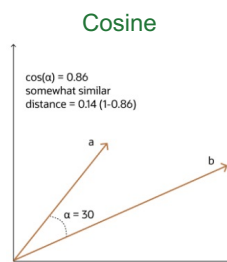
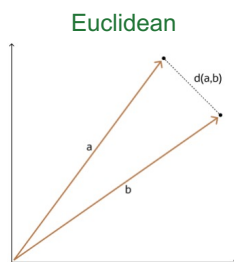
VECTOR_DISTANCE calculates the distance between two vectors

vector_distance (vector1, vector2, method)

returns a number containing the distance between vectors 1 and 2, calculated according to the method specified in the 3rd parameter.

88

Vector distance measurement methods



A	1	0	0	1	0	1	1	0
B	1	1	0	1	1	0	1	0
XOR Bit Operations								
	0	1	0	0	1	1	0	0

Hamming Distance = 3

Hamming

89

VECTOR_DISTANCE – Example

```
select vector_distance
      (s.the_vector, P_SEARCH_VECTOR, COSINE) as vdist,
      s.the_text,
      s.id
  from sentences s
 order by vdist asc
fetch first 8 rows only;
```

90

Creating a Text Search Configuration – AI Vector Search

Create Search Configuration

Detail

A **Search Configuration** contains information about the searchable *Data Source*, the functionality used for search, as well as display properties.

* Name ?

* Search Type

- ☐ Standard ?
- ☒ **Oracle Vector Search**
- ☐ Oracle Text
- ☐ Oracle Ubiquitous Search
- ☐ List

Cancel Next >

91

Creating a Text Search Configuration – AI Vector Search

Create Search Configuration

✓

●

Source

* Vector Provider ?

Source Type **Table** SQL Query ?

Table / View Owner ?

* Table / View Name ?

< Cancel Next >

92

Creating a Text Search Configuration – AI Vector Search

Create Search Configuration

✓

✓

●

Column Mapping

* Primary Key Column ?

* Vector Column ?

* Title Column ?

Description Column ?

Icon Source **Initials** Icon Class Icon Class Column ?

< Cancel Create Search Configuration

93

Creating a Text Search Configuration – AI Vector Search - Attributes

Source

Source Type: **Table** SQL Query ?

Table / View Owner: - Parsing Schema - ?

* Table / View Name: ?

Where Clause: ?

Vector Attributes

Provider: Inside DB ?

* Column Name: THE_VECTOR (Vector) ?

Search Type: Exact ?

Distance Metric: Cosine ?

94

Creating a Text Search Configuration – AI Vector Search - Attributes

Source

Source Type: **Table** SQL Query ?

Table / View Owner: - Parsing Schema - ?

* Table / View Name: ?

Where Clause: ?

Vector Attributes

Provider: Inside DB ?

* Column Name: THE_VECTOR (Vector) ?

Search Type: Exact ?

Distance Metric: Cosine ?

95

AI Vector Search

APEX search configs

Standard Oracle Text (Search Index) AI Vector

AI Vector sentences

building

AK Astana, Kazakhstan is admired for its futuristic skyline and modernist buildings.

SN Skopje, North Macedonia is noted for its eclectic blend of old and new buildings.

TA Tirana, Albania is known for its colorful buildings and vibrant cultural life.

TG Tbilisi, Georgia is celebrated for its unique blend of old and new architecture.

MB Minsk, Belarus is recognized for its Soviet-era architecture and green spaces.

96

AI Vector Search

Generated Component SQL Query:

```
select i.*
  from (select i.*,(vector_distance( THE_VECTOR, :apex$f1, COSINE )) "APEX$VECTOR_DISTANCE"
  from(select /*+ qb_name(apex$inner) */* from(select x.* from "SENTENCES2" x
)d
)i
)i where 1=1
order by "APEX$VECTOR_DISTANCE" fetch exact first 100 rows only
```

97

AI Vector Search

Demo



98

AI Vector Hybrid Search



99

Wouldn't it be nice if we could mix both Oracle Text Search and AI Vector Search together?

100

AI Vector Hybrid Search

With AI Vector Hybrid Search we can retrieve both Oracle Text and AI Vector Search results with a single query

101

AI Vector Hybrid Search

- Creating a hybrid vector index

```
create hybrid vector index
    sentences_hybrid_idx
on sentences(the_text)
    parameters ('model all_minilm_l12_v2')
parallel 8;
```



102

AI Vector Hybrid Search - query

- Search using the newly created index

```
select json_serialize(
    dbms_hybrid_vector.search(
        json(
            '{
                "hybrid_index_name" : "sentences_hybrid_idx",
                "search_text"       : "river"
            }'
        )
    )
    returning clob pretty) as output;
```



103

AI Vector Hybrid Search – query result

```
[
  {
    "rowid" : "AAArS1AAAAAGAL+ABJ",
    "score" : 56.97,
    "vector_score" : 61.67,
    "text_score" : 10,
    "vector_rank" : 1,
    "text_rank" : 1,
    "chunk_text" : "Budapest, Hungary is notable for its beautiful Danube River
and thermal baths.",
    "chunk_id" : "1"
  },
  {
    "rowid" : "AAArS1AAAAAGAL/AA0",
    "score" : 53.58,
    "vector_score" : 58.94,
    "text_score" : 0,
    "vector_rank" : 2,
    "text_rank" : 80,
    "chunk_text" : "Gothenburg, Sweden is recognized for its Dutch-style canals
and cultural events.",
    "chunk_id" : "1"
  },
  {
    "rowid" : "AAArS1AAAAAGAL/AA5",
    "score" : 53.16
```

104

AI Vector Hybrid Search – query with json_table

```
select *
  from json_table(dbms_hybrid_vector.search(
    json('{"hybrid_index_name" : "sentences_hybrid_idx",
          "search_text" : "wool" }'),
    )
    , '$[*]'
    columns (score          varchar2(20) path '$.score',
             vector_score   varchar2(20) path '$.vector_score',
             text_score     varchar2(20) path '$.text_score',
             vector_rank    varchar2(20) path '$.vector_rank',
             text_rank      varchar2(20) path '$.text_rank',
             chunk_text     varchar2(200) path '$.chunk_text',
             chunk_id       varchar2(20) path '$.chunk_id') );
```

105

AI Vector Hybrid Search – query with json_table

SCORE	VECTOR_SCORE	TEXT_SCORE	VECTOR_RANK	TEXT_RANK	CHUNK_TEXT
66.05	71.85	8	1	1	Sheep are valued for their wool, meat, and milk.
63.57	69.13	8	2	1	Alpacas are kept for their soft, luxurious wool.
61.95	67.35	8	3	1	Llamas are used as pack animals and for their wool.
53.51	58.86	0	4	80	Sheepdogs are specifically bred and trained to help herd sheep
52.1	57.31	0	5	80	Cows are essential farm animals, providing milk, meat, and tea
51.61	56.77	0	6	80	Goats are known for their versatility, providing milk, meat, a
51.49	56.64	0	7	80	Rabbits are sometimes kept on farms for their meat and fur.
51.09	56.2	0	8	80	Emus are large birds farmed for their meat, oil, and leather.
51.01	56.11	0	9	80	Hedgehogs are admired for their unique appearance and quiet na
50.65	55.71	0	10	80	Rust is a municipality in the district of Ortenau in Baden-Wür
50.28	55.31	0	11	80	Nuuk, Greenland is recognized for its stunning arctic landscap
50.26	55.29	0	12	80	Dogs are often used on farms to herd livestock and provide pro
50.09	55.1	0	13	80	Tarantulas are kept by enthusiasts for their unique appearance
49.96	54.96	0	14	80	Peacocks are kept on farms and estates for their ornamental be
49.93	54.92	0	15	80	Cats are known for their ability to control rodent populations
49.85	54.83	0	16	80	Copenhagen, Denmark is famous for its colorful Nyhavn harbor a
49.75	54.72	0	17	80	Horses are essential in many agricultural tasks, from plowing
49.6	54.56	0	18	80	Malmo, Sweden is known for its modern architecture and green s
49.44	54.38	0	19	80	Alpacas are also kept as pets and for their calm and friendly
49.25	54.17	0	20	80	Horses are prized for their strength, speed, and companionship

106

AI Vector Hybrid Search

Demo

107



108

APEX Search Configurations - what we've seen...

Standard search

Oracle Text (Basic, Search Engine and Expert)

Ubiquitous Search

AI Vector Search

AI Hybrid Vector Search



109

