

Oracle SQL Developer & REST Data Services

What's New

Jeff Smith
Senior Principal Product Manager
Database Development Tools
Jeff.d.smith@oracle.com
@thatjeffsmith
<http://www.thatjeffsmith.com>



Agenda

- New & Improved Features
- Slides + Live Demo
- Questions? Please!
- Let's Go!

Improve
application
developers
experience



Safe Harbor Statement

The preceding 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.

Oracle SQL Developer, Major Feature Areas

- PL/SQL IDE
- SQL editor
- GUI for browsing and managing database objects
- Ad Hoc reporting
- Database Design & Data Modeling
- Database Administration
- 3rd Party RDBMS Migrations to Oracle
- Deploy and Administer Oracle REST Data Services
- REST Enablement of the Database
- Command Line Interface



Remember Last Year...

...we debuted Oracle SQLcl!

```
185 Alexis Bull ABULL 650.509.2876 20-FEB-997 12:00:00 SH_CLERK 4100
186 Julia Dellinger JDELLING 650.509.3876 24-JUN-998 12:00:00 SH_CLERK 3400
20 SMAVRIS 187 Anthony Cabrio ACABRIO 650.509.4876 07-FEB-999 12:00:00 SH_CLERK 3000
188 Kelly Chung KCHUNG 650.505.1876 14-JUN-997 12:00:00 SH_CLERK 3800
EMPLOYEE_ID 189 Jennifer Dilly JDILLY 650.505.2876 13-AUG-997 12:00:00 SH_CLERK 3600
----- 190 Timothy Gates TGATES 650.505.3876 11-JUL-998 12:00:00 SH_CLERK 2900
EMAIL 191 Randall Perkins RPERKINS 650.505.4876 19-DEC-999 12:00:00 SH_CLERK 2500
----- 192 Sarah Bell SBELL 650.501.1876 04-FEB-996 12:00:00 SH_CLERK 4000
SALARY 193 Britney Everett BEVERETT 650.501.2876 03-MAR-997 12:00:00 SH_CLERK 3900
----- 194 Samuel McCain SMCCAIN 650.501.3876 01-JUL-998 12:00:00 SH_CLERK 3200
6500 195 Vance Jones VJONES 650.501.4876 17-MAR-999 12:00:00 SH_CLERK 2800
20 196 Alana Walsh AWALSH 650.507.9811 24-APR-998 12:00:00 SH_CLERK 3100
HBAER 197 Kevin Feeney KFEENEY 650.507.9822 23-MAY-998 12:00:00 SH_CLERK 3000
10000 198 Donald OConnell DOCONNEL 650.507.9833 21-JUN-999 12:00:00 SH_CLERK 2600
20 199 Douglas Grant DGRANT 650.507.9844 13-JAN-000 12:00:00 SH_CLERK 2600
SHIGGINS
12000 EMPLOYEE_ID FIRST_NAME LAST_NAME EMAIL PHONE_NUMBER HIRE_DATE JOB_ID SALARY COMMISSION
200 Jennifer Whalen JWHALEN 515.123.4444 17-SEP-987 12:00:00 AD_ASST 4000
201 Michael Hartstein MHARTSTE 515.123.5555 17-FEB-996 12:00:00 MK_MAN 13000
WGIETZ 202 Pat Fay PFAY 603.123.6666 17-AUG-997 12:00:00 MK_REP 6000
8300 203 Susan Mavris SMAVRIS 515.123.7777 07-JUN-994 12:00:00 HR_REP 6500
204 Hermann Baer HBAER 515.123.8888 07-JUN-994 12:00:00 PR_REP 10000
107 rows s 205 Shelley Higgins SHIGGINS 515.123.8080 07-JUN-994 12:00:00 AC_MGR 12000
206 William Gietz WGIETZ 515.123.8181 07-JUN-994 12:00:00 AC_ACCOUNT 8300

YUCK!
107 rows selected.
```

YEAH!

SQLcl: a modern take on SQL*Plus

Included with SQL Developer, also available as a separate download/program

- Java 8 (JRE)
- < 20 MB
- Unzip & Go
- No Client Required







```
SQL> info sh.sales
TABLE: SALES
      LAST ANALYZED:2014-12-13 06:01:41.0
      ROWS           :918843
      SAMPLE SIZE    :918843
      INMEMORY       :
      COMMENTS       :facts table, without a primary key; all rows are uniquely identified by the combination of all foreign keys

Columns
NAME      DATA TYPE  NULL  DEFAULT  COMMENTS
-----
PROD_ID   NUMBER      No              FK to the products dimension table
CUST_ID   NUMBER      No              FK to the customers dimension table
TIME_ID   DATE        No              FK to the times dimension table
CHANNEL_ID NUMBER      No              FK to the channels dimension table
PROMO_ID  NUMBER      No              promotion identifier, without FK constraint (intentionally) to show outer join optimization
QUANTITY_SOLD NUMBER(10,2) No              product quantity sold with the transaction
AMOUNT_SOLD NUMBER(10,2) No              invoiced amount to the customer

Indexes
INDEX NAME      UNIQUENESS  STATUS  FUNCIDX STATUS  COLUMNS      COLUMN EXPRESSION
-----
SH.SALES_CUST_BIX  NONUNIQUE  N/A          CUST_ID
SH.SALES_PROD_BIX  NONUNIQUE  N/A          PROD_ID
SH.SALES_TIME_BIX  NONUNIQUE  N/A          TIME_ID
SH.SALES_PROMO_BIX  NONUNIQUE  N/A          PROMO_ID
SH.SALES_CHANNEL_BIX  NONUNIQUE  N/A          CHANNEL_ID

SQL>
```

...We Could Take the Helpful GUI Elements to a CLI?

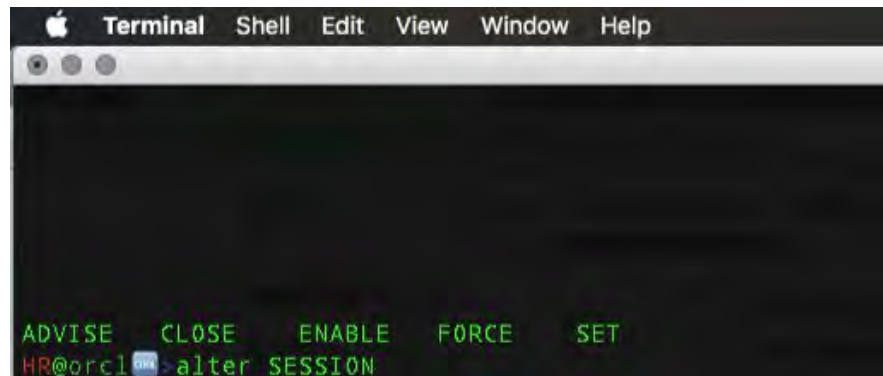
- Object Name/Command Completion 
- SQL Execution History & Recall 
- Query Result Formatting 
- Modern Editor 
- Custom Commands 
- Client-Side JavaScripting 
- All the AWESOME!!!



Insight/Auto-Complete Object Names and Commands

```
CLOB_STUFF      COMMISSION_PCT  DEPARTMENT_ID  EMAIL          EMPLOYEE_ID
EMP_DEVICE      FIRST_NAME      HIRE_DATE      JOB_ID         LAST_NAME
MANAGER_ID      PHONE_NUMBER    REVIEW         SALARY
HR@orcl >select
  2  from hr.employees
  3  where COMMISSION_PCT      is not null
```

- Just hit <TAB> for help
- Autocompletes on Single Match
- Lists possible matches otherwise!

A screenshot of a macOS Terminal window titled "Terminal" with menu items "Shell", "Edit", "View", "Window", and "Help". The terminal shows a SQL command being entered: "HR@orcl >alter SESSION". The word "SESSION" is highlighted in green, and a list of possible completions is shown above it: "ADVISE", "CLOSE", "ENABLE", "FORCE", and "SET".

```
Terminal  Shell  Edit  View  Window  Help
HR@orcl >alter SESSION
ADVISE  CLOSE  ENABLE  FORCE  SET
```

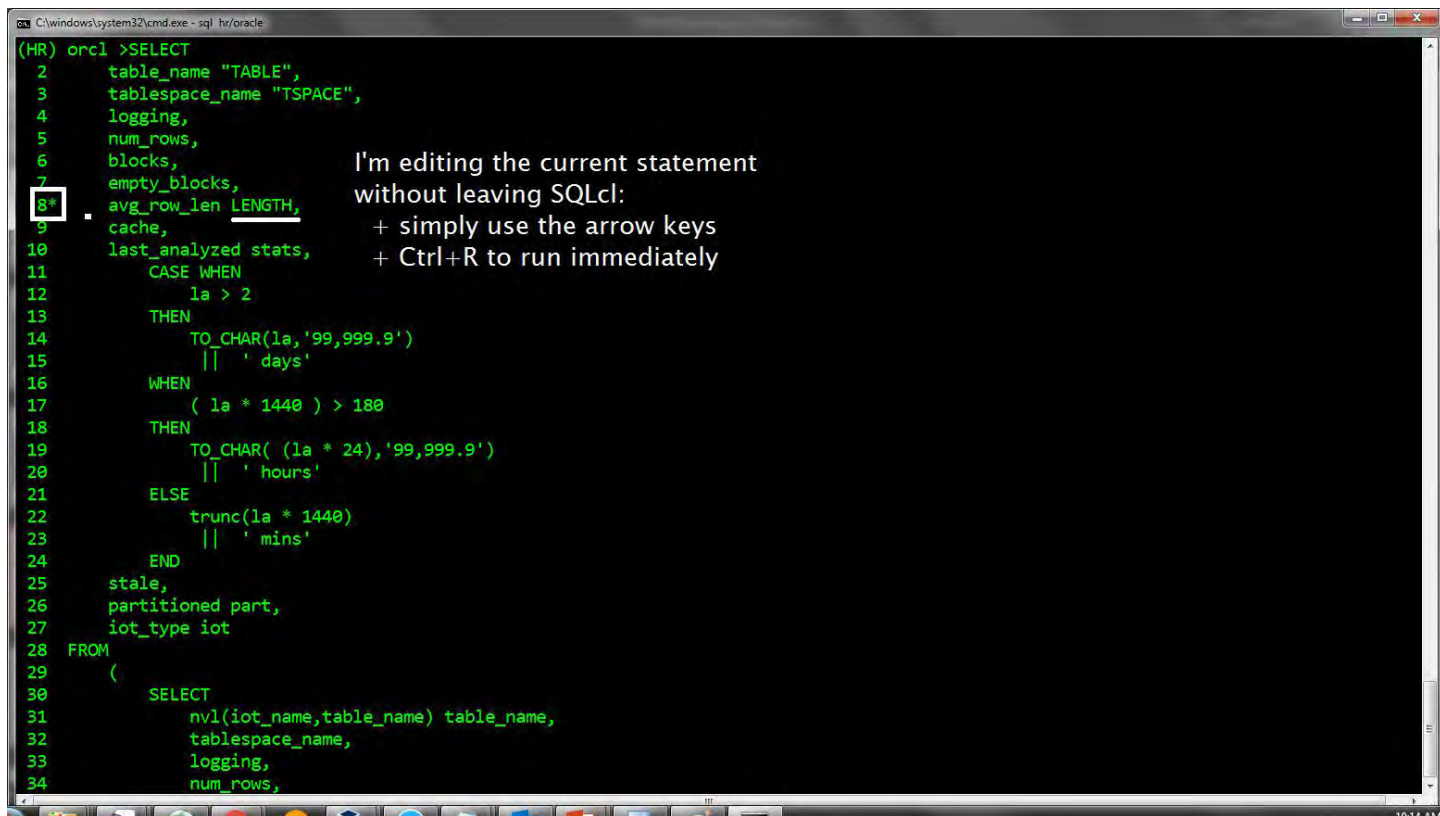

SQL Statement & Script History

```
Terminal Shell Edit View Window Help
bin
HR@orcl >history full
 1 info EVERYFIRSTANDTHIRDWED
 2 ddl EVERYFIRSTANDTHIRDWED
 3 alias dom=SELECT
  > table_name "TABLE",
  > tablespace_name "TABLESPACE",
  > logging,
  > num_rows,
  > blocks,
  > empty_blocks,
  > avg_row_len,
  > cache,
  > last_analyzed stats,
  > CASE WHEN
  > 1a > 2
  > THEN
```

```
Terminal Shell Edit View Window Help
bin — java + sql hr/oracle
HR@orcl >history
History:
 1 info EVERYFIRSTANDTHIRDWED
 2 ddl EVERYFIRSTANDTHIRDWED
 3 alias dom=SELECT table_name "TABLE",
 4 show sqlformat
 5 select count(*) from v$session where username
 6 select * from sh.sales
 7 connect sys as sysdba
 8 alter user sh unlocked
 9 alter user sh status unlocked
10 connect sh/oracle
11 alias list dom
12 tables2
13 host cat login.sql
14 host cat login.js
```

```
Terminal Shell Edit View Window Help
bin — java + sql hr/oracle — S
HR@orcl >history 69
 1 select FIRST_NAME, LAST_NAME, SALARY, HIRE_DATE
 2 from hr.employees
 3* where COMMISSION_PCT is not null
HR@orcl >
```

The In-Line Editor



```
(HR) orcl >SELECT
 2  table_name "TABLE",
 3  tablespace_name "TSPACE",
 4  logging,
 5  num_rows,
 6  blocks,
 7  empty_blocks,
 8* . avg_row_len LENGTH,
 9  cache,
10  last_analyzed stats,
11  CASE WHEN
12    la > 2
13  THEN
14    TO_CHAR(la, '99,999.9')
15    || ' days'
16  WHEN
17    ( la * 1440 ) > 180
18  THEN
19    TO_CHAR( (la * 24), '99,999.9')
20    || ' hours'
21  ELSE
22    trunc(la * 1440)
23    || ' mins'
24  END
25  stale,
26  partitioned part,
27  iot_type iot
28 FROM
29  (
30  SELECT
31    nvl(iot_name,table_name) table_name,
32    tablespace_name,
33    logging,
34    num_rows,
```

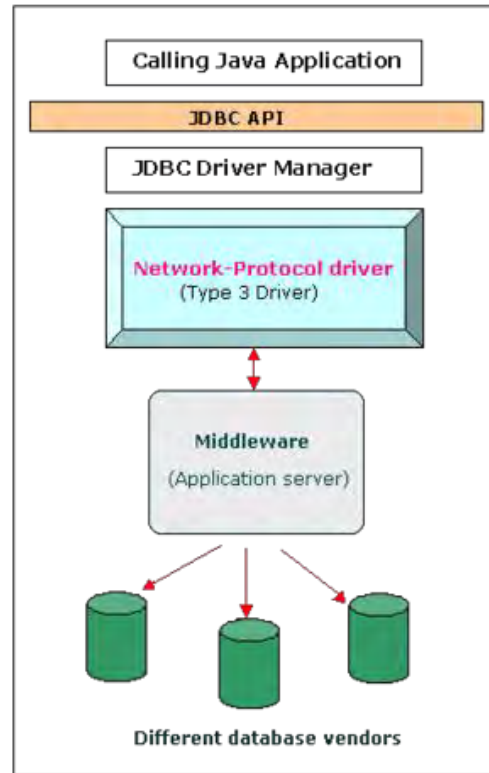
I'm editing the current statement
without leaving SQLcl:
+ simply use the arrow keys
+ Ctrl+R to run immediately

What's New?

- Ships with a new JDBC Driver
 - Connect to DB via ORDS
 - Works over REST...to a new `_/sql/` service
 - DB Auth
 - Stateless
 - Opens up access to...

New REST-based ORDS Powered JDBC Driver

- type 3 driver
- Middleware = ORDS
- JAR avail on OTN



No Database Access? No Problem??

```
I am on > connect HR/oracle@jdbc:oracle:orest:@http://localhost:8888/ords/hr/  
Connected.
```

```
Session altered.
```

```
DDL Option PRETTY on  
DDL Option CONSTRAINTS off  
DDL Option REF_CONSTRAINTS off  
DDL Option PARTITIONING on  
DDL Option TABLESPACE on  
DDL Option SEGMENT_ATTRIBUTES off  
DDL Option STORAGE off
```

```
'HELLOJEFF'
```

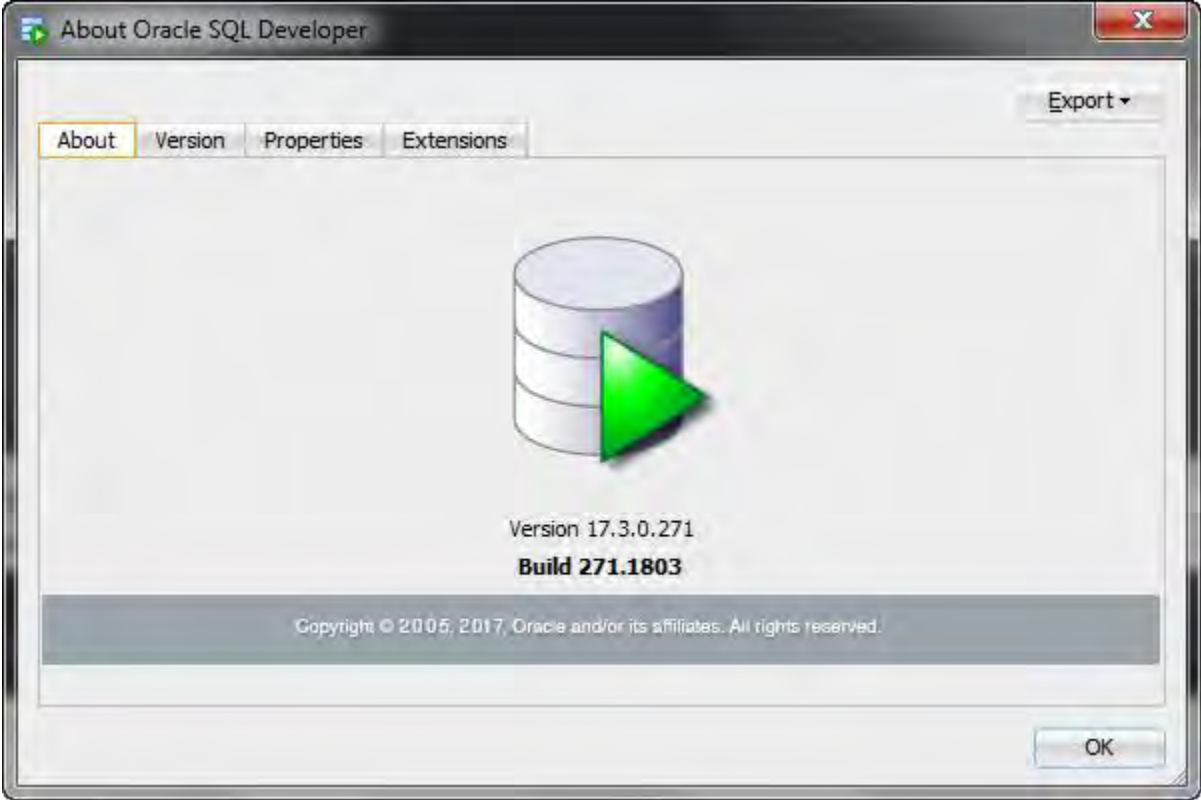
```
hello jeff
```

```
I am HR on jdbc:oracle:orest:@http://localhost:8888/ords/hr/ > _
```



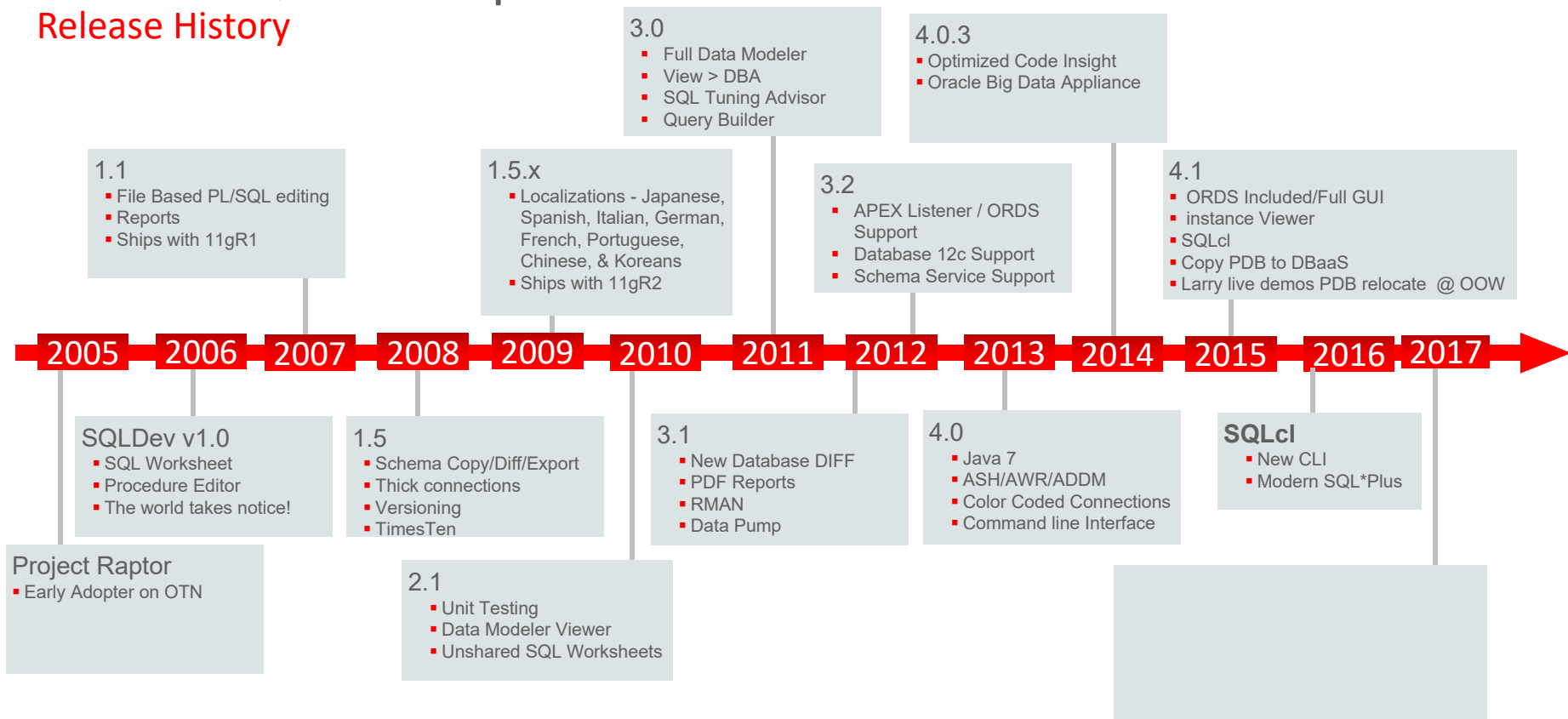
Oracle REST Data Services

Oracle SQL Developer, New Tricks Up Our Sleeve?



Oracle SQL Developer

Release History



Oracle SQL Developer

Release **FUTURE**

2017.2

- Bug Fixes
- Sharding
- Data Guard

17.3

(Coming Soon/Now Available)

- Formatter/Parser Tweaks
- Bug Fixes

2017

2018

17.4/18.1

- *SQL Developer WEB*
- *Connectivity Enhancements*
- *Oracle Cloud Services Data Loader (Cart)*

Quarterly Releases!!!



Start Page | Sharded DB Viewer | 2d | Databases

east *west*

gsm1

shd1 *dbs1* *sqldev2b* *sqldev2c* *dbs2* *sqldev2d* *sqldev2e*

Database: UP
Chunks: 2
Broker Set: 1

Database: UP
Chunks: 2
Broker Set: 2

Database: UP
Chunks: 2
Broker Set: 1

Database: UP
Chunks: 2
Broker Set: 2

**Shard sqldev2b PRIMARY
Broker Set 1 is highlighted
Associated Connection is 2b**

- Open Instance Viewer
- Open Data Guard Console**
- Edit Connection Association...
- Remove Connection Association

Oracle SQL Developer: DATAGUARD SYS null@12c

File Edit View Navigate Run Team Tools Window Help Automation

Connections

Start Page Sharded DB Viewer 20 Databases Console

Local Values

Properties Statistics Messages

Name	Value
1 Database Role	PHYSICAL STANDBY
2 Switchover Status	NOT ALLOWED
3 Dataguard Broker	ENABLED
4 Guard Status	NONE

Properties Statistics Messages

FACILITY	SEVERITY	DEST_ID	MESSAGE
1 Log Transport Services	Control	0	13
2 Log Apply Services	Warning	0	13
3 Log Transport Services	Informational	1	13
4 Log Transport Services	Control	1	13
5 Remote File Server	Informational	0	13
6 Remote File Server	Warning	0	13
7 Remote File Server	Informational	0	13
8 Remote File Server	Control	0	13
9 Log Transport Services	Warning	0	13
10 Log Transport Services	Informational	0	13
11 Log Transport Services	Control	1	13
12 Log Transport Services	Control	0	13

Properties Statistics Messages

NAME	VALUE	UNIT
1 transport lag	+00 00:00:00	day(2)
2 apply lag	+00 00:00:00	day(2)
3 apply finish time	+00 00:00:00.000	day(2)
4 estimated startup time	13	second

Connections

Sharding

2b
2c
2d
2e
Sharding

DBA

Connections

2e

Database Configuration

Database Status

Dataguard

Console

Databases

sqldev2c
sqldev2n

Local Values

Data Pump

Performance

RMAN Backup/Recovery

Resource Manager

SQL Translator Framework

Scheduler

Security

Storage

Tuning

exa express JEFF

hr

Sharding

Database Configuration

Database Status

Data Pump

Oracle Sharding

Sharded DB Viewer

sqldev2c

Role Status PRIMARY 0

sqldev2e

Role Status STANDBY 10019

Cloud Computing

ORACLE SQL Developer 17.3



- **FREE** Oracle Database IDE/GUI
- Windows, OS X, *NIX
- More than 4,500,000 users worldwide
- My Oracle Support available via your DB license

Database Search Improved: Source Code

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Code' section of the search results is expanded, and the 'ORDS' checkbox is checked and highlighted with a red box. The main pane shows a tree view of search results for 'HR' objects, including 'emps->otn', 'items->lineitems', and 'plsq_offset/->examples'. A red arrow points from the text 'Click to open' to the 'GET emps' worksheet tab. The worksheet tab is active, showing a SQL query in a dark-themed editor:

```
1 begin
2   get_emp_json(
3     p_empid => :p_emp_id
4   );
5 end;
```

- REST Handlers
- Views
- Materialized Views
- Operators

Because copying code from the Internet should be easy 😊

The image illustrates a common mistake when copying code from the Internet. It shows three stages of an Oracle SQL Developer window:

- Top Screenshot:** A complex SQL query is displayed. A tooltip indicates an "ASCII symbol out of range" error on line 4, which is the first line of the copied code.
- Middle Screenshot:** The same query is shown. A tooltip indicates "Copied line numbers" and a "Fix it" button, suggesting the user is trying to correct the error by adjusting line numbers.
- Bottom Screenshot:** The query is now simplified to `SELECT * FROM NOT A TABLE OR VIEW;`. A red dashed underline is under the text, and a tooltip indicates "Undefined Table", showing the result of a failed copy-paste operation.

More Formatter Improvements

The image shows a screenshot of the Oracle SQL Developer interface. On the left, the 'Worksheet' tab displays a SQL query with line numbers 199 through 225. The query is a complex SELECT statement with multiple subqueries and joins. On the right, the 'Preferences' dialog is open, showing the 'Format: Advanced Format' section. The 'Advanced Format' section is expanded, and the 'Line Breaks' sub-section is visible. The 'Max char line width' is set to 128. Other options like 'On comma', 'On procedure arguments', and 'On concatenation' are set to 'After', while 'On Boolean connectors' and 'On ANSI joins' are set to 'Before'. The 'On subqueries' option is checked. The 'Named Argument Separator' and 'Assignment Operator :=' options are also checked. The 'Indentation' section shows 'Indent spaces' set to 4. The 'White Space' section is partially visible at the bottom.

```
199      "A17"."RELFILE#" "RELATIVE_FNO",
200      DECODE("A16"."KTFBHCCVAL",0,DECODE("A16"."KTFBHCINC",0,'NO','YES'),NULL) "AUTOEXTENSIBLE",
201      DECODE("A16"."KTFBHCCVAL",0,"A15"."BLOCKSIZE" * "A16"."KTFBHCMAXSZ" NULL) "MAXBYTES",
202      DECODE("A16"."KTFBHCCVAL",0,"A16"."KTFBHCMAXSZ",N
203      DECODE("A16"."KTFBHCCVAL",0,"A16"."KTFBHCINC",N
204      DECODE("A16"."KTFBHCCVAL",0,"A16"."KTFBHCUSZ" *
205      DECODE("A16"."KTFBHCCVAL",0,"A16"."KTFBHCUSZ",N
206      DECODE("A14"."FETSN",0,DECODE(bitand("A14"."FES
207      ,'RECOVER') ) "ONLINE_STATUS",
208      DECODE(bitand("A17"."SPARE2",3),NULL,'OFF',0,'O
209      FROM (
210      SELECT "A3"."FILE#" "FILE#",
211             "A3"."NAME" "NAME",
212             "A3"."CON_ID" "CON_ID"
213      FROM (
214      SELECT "A4"."FILE#" "FILE#",
215             "A4"."NAME" "NAME",
216             "A4"."CON_ID" "CON_ID"
217      FROM (
218      SELECT "A5"."INST_ID" "INST_ID",
219             "A5"."FNFNO" "FILE#",
220             "A5"."FNNAM" "NAME",
221             "A5"."CON_ID" "CON_ID"
222      FROM "SYS"."X$KCCFN" "A5"
223      WHERE "A5"."FNNAM" IS NOT NULL
224      AND bitand("A5"."FNFLG",4) <> 4
225      AND "A5"."ENTY2" = 4
```

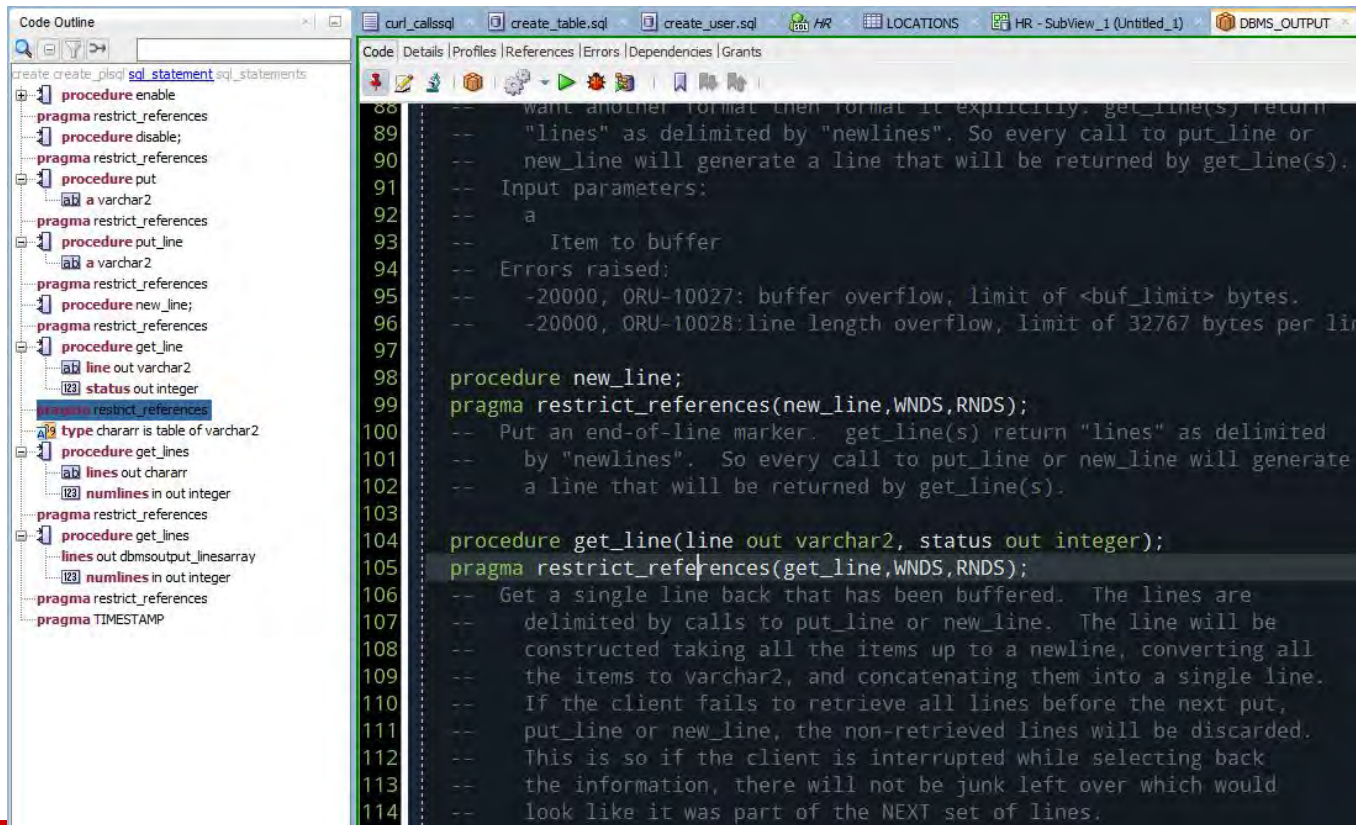
Preferences

Format: Advanced Format

- Named Argument Separator =>
- Assignment Operator :=
- Equality Predicate =
- Indentation
 - Indent spaces Spaces
 - Indent with
- Line Breaks
 - On comma
 - On procedure arguments
 - On concatenation
 - On Boolean connectors
 - On ANSI joins
 - For compound_condition parenthesis
 - On subqueries
 - Max char line width
 - Before line comments
 - After statements
 - SELECT/FROM/WHERE
- CASE
- IF
- LOOP
- White Space

Import... Export... Reset

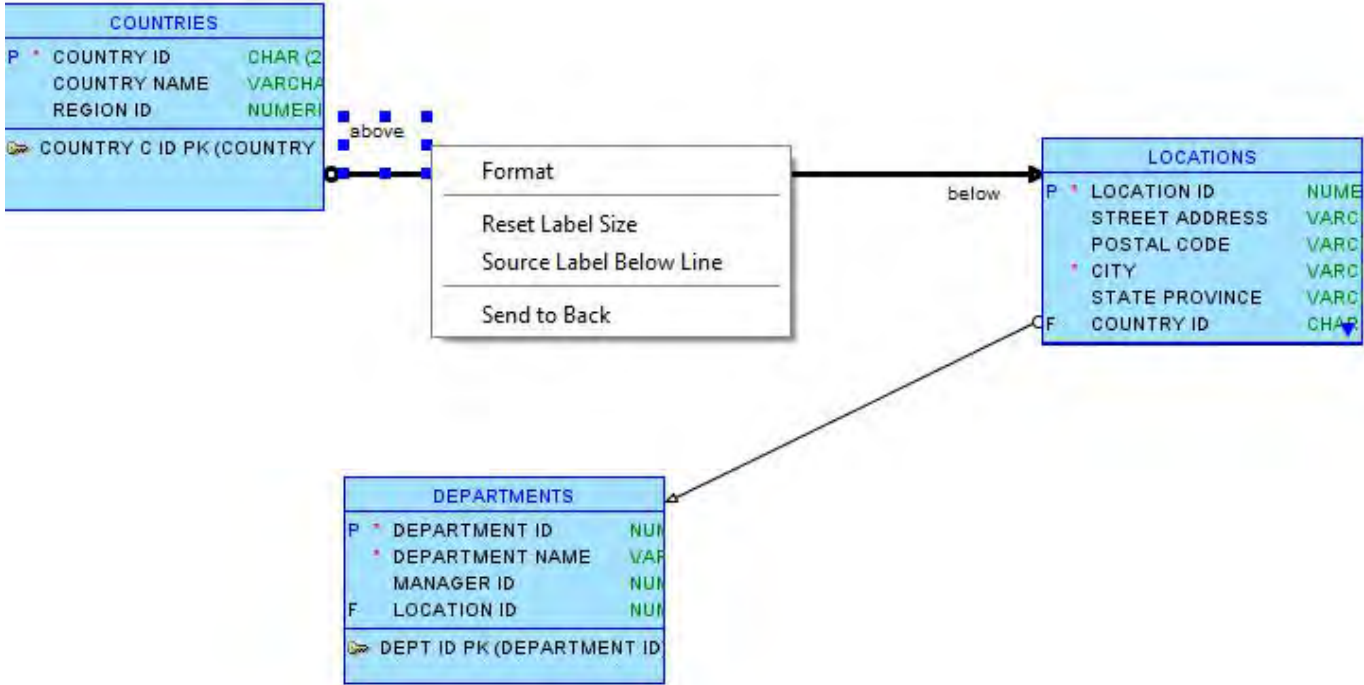
Code Outline too!



The screenshot displays the Oracle SQL Developer interface. On the left, the 'Code Outline' pane shows a tree view of the code structure. The main editor on the right shows the source code for a PL/SQL package named 'dbms_output'. The code includes comments explaining the behavior of the 'get_line(s)' procedure, which returns lines as delimited by 'newlines'. The code also defines procedures for 'enable', 'disable', 'put', 'put_line', 'new_line', and 'get_line', along with their respective parameters and error handling.

```
88 -- want another format then format it explicitly. get_line(s) return
89 -- "lines" as delimited by "newlines". So every call to put_line or
90 -- new_line will generate a line that will be returned by get_line(s).
91 -- Input parameters:
92 --   a
93 --   Item to buffer
94 -- Errors raised:
95 --   -20000, ORU-10027: buffer overflow, limit of <buf_limit> bytes.
96 --   -20000, ORU-10028: line length overflow, limit of 32767 bytes per lin
97
98 procedure new_line;
99 pragma restrict_references(new_line,WNDS,RNDS);
100 -- Put an end-of-line marker. get_line(s) return "lines" as delimited
101 -- by "newlines". So every call to put_line or new_line will generate
102 -- a line that will be returned by get_line(s).
103
104 procedure get_line(line out varchar2, status out integer);
105 pragma restrict_references(get_line,WNDS,RNDS);
106 -- Get a single line back that has been buffered. The lines are
107 -- delimited by calls to put_line or new_line. The line will be
108 -- constructed taking all the items up to a newline, converting all
109 -- the items to varchar2, and concatenating them into a single line.
110 -- If the client fails to retrieve all lines before the next put,
111 -- put_line or new_line, the non-retrieved lines will be discarded.
112 -- This is so if the client is interrupted while selecting back
113 -- the information, there will not be junk left over which would
114 -- look like it was part of the NEXT set of lines.
```

ER Diagram Relationship Labels



Coming Soon...17.4/18.1

- Cart, Data Loading to Oracle DB Cloud Services
 - OSS Support
- Amazon RedShift migration support
- Easier Connection management
- SQL Developer Web – Full Demo – In my NEXT Talk 😊

✓ Oracle SQL Developer: GUI, CLI, or Browser? [CON6670]

This session provides an overview of Oracle's current development tools. Learn how to get started, see basic features for the developer, and learn how they work, plus see lots of demos. This session also showcases the new web implementation of Oracle SQL Developer. As a bonus, attendees get new tips and tricks to make the database development experience that much better.

Jeff Smith, Senior Principal Product Manager, Oracle

⊕ **Monday, Oct 02**, 3:15 p.m. - 4:00 p.m. | Moscone West - Room 3012

Oracle REST Data Services



Formerly Known as the APEX Listener

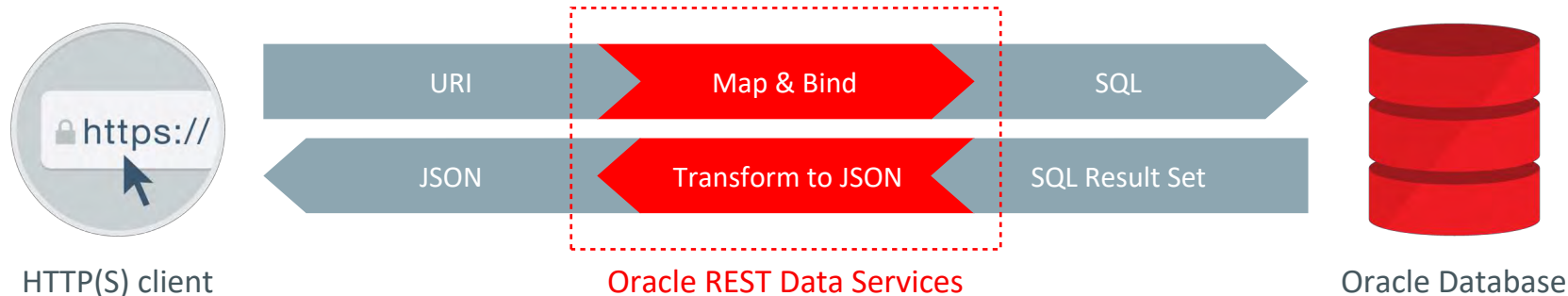
REpresentational State Transfer (REST)

It relies on a stateless, client-server, cacheable communications protocol -- and in virtually all cases, the HTTP(S!) protocol is used.

REST is *an architecture style* for designing networked applications. The idea is that, rather than using complex mechanisms such as CORBA, RPC or SOAP to connect between machines, simple HTTP is used to make calls between machines. (rest.elkstein.org)

- **REST is easy**
 - Small uniform set of operations: GET, POST, PUT, DELETE (CRUD!)
 - Small set of uniform status codes: 200, 201, 404, 500, ...
 - URLs & hyperlinks encourage stateless behavior
 - Text based protocol with simple request/response model

So How do we Marry this Cool, 'New', Hip Thing to the DB?

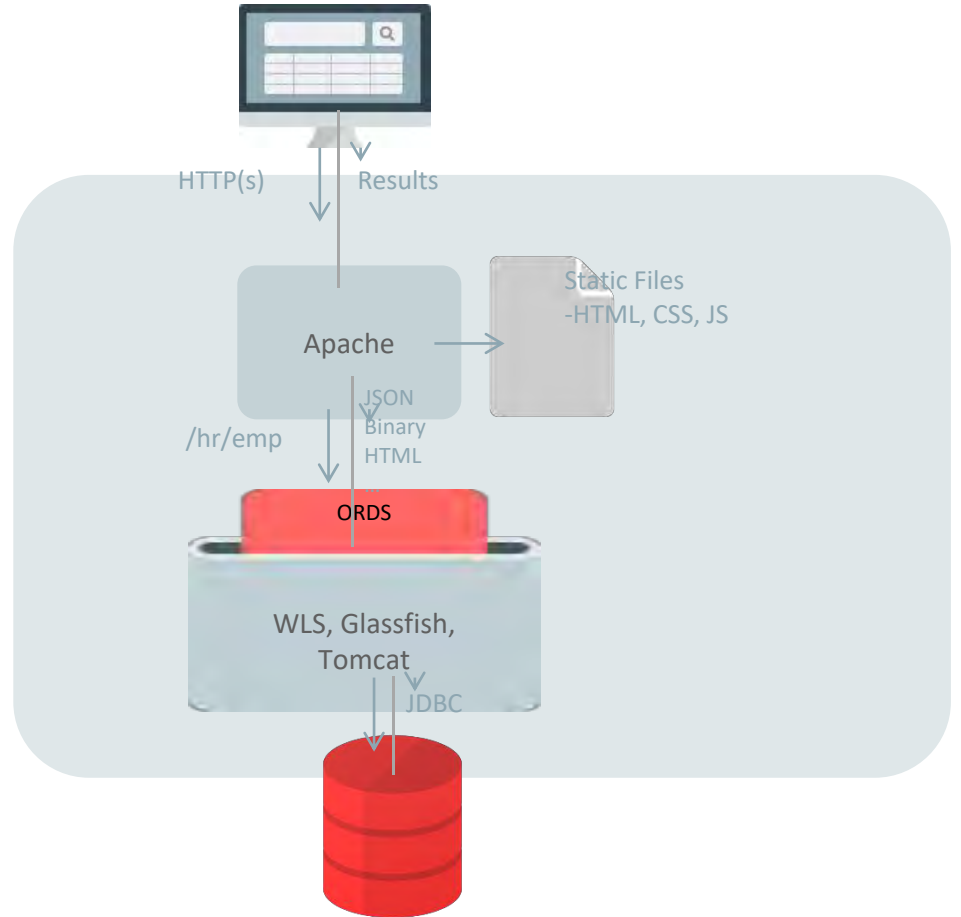


- Java JEE mid tier application, e.g., WebLogic, Tomcat, Glassfish
 - Also supports “Standalone” mode for development
- For input, maps/binds URI to SQL and PL/SQL
- For output, transforms results to JSON and other formats

Oracle REST Data Services

Flexible architecture

- Standard webserver layout
- Implements Java Servlet
- Deploys to WLS, Glassfish, Tomcat
- OR Embedded Jetty for standalone operation



And it's Easy!

I had been looking for a chance to perform a POC, so I proposed I just provide him with a couple of web services.

In under a day I had functioning web services in place for him to consume.

There's been refinements to them and new ones developed since then. We are now exploring where else we could leverage this technology.

- customer talking about ORDS

What's It Get Me?

The screenshot shows an Oracle APEX REST client interface for a REST service named "Image Gallery Post a File". The request method is "POST" and the URL is "http://localhost:8888/ords/hr/test2/media". The response body is displayed in JSON format:

```
1 - {  
2   "X-APEX-FORWARD": "1",  
3   "X-APEX-STATUS-CODE": 201  
4 }
```

The "Choose Files" button is highlighted with a red box, and the file "ctrl-click-nav.gif" is selected. The SQL Developer window shows the following SQL script:

```
1 declare «  
2   image_id integer; «  
3 begin «  
4   insert into gallery (title, content_type, image) «  
5     ..... values (:title, :content_type, :body) «  
6     ..... returning id into image_id; «  
7   :status := 201; «  
8   :LOCATION := IMAGE_ID; «  
9 end;
```

The SQL Developer window also shows the "GALLERY" table structure:

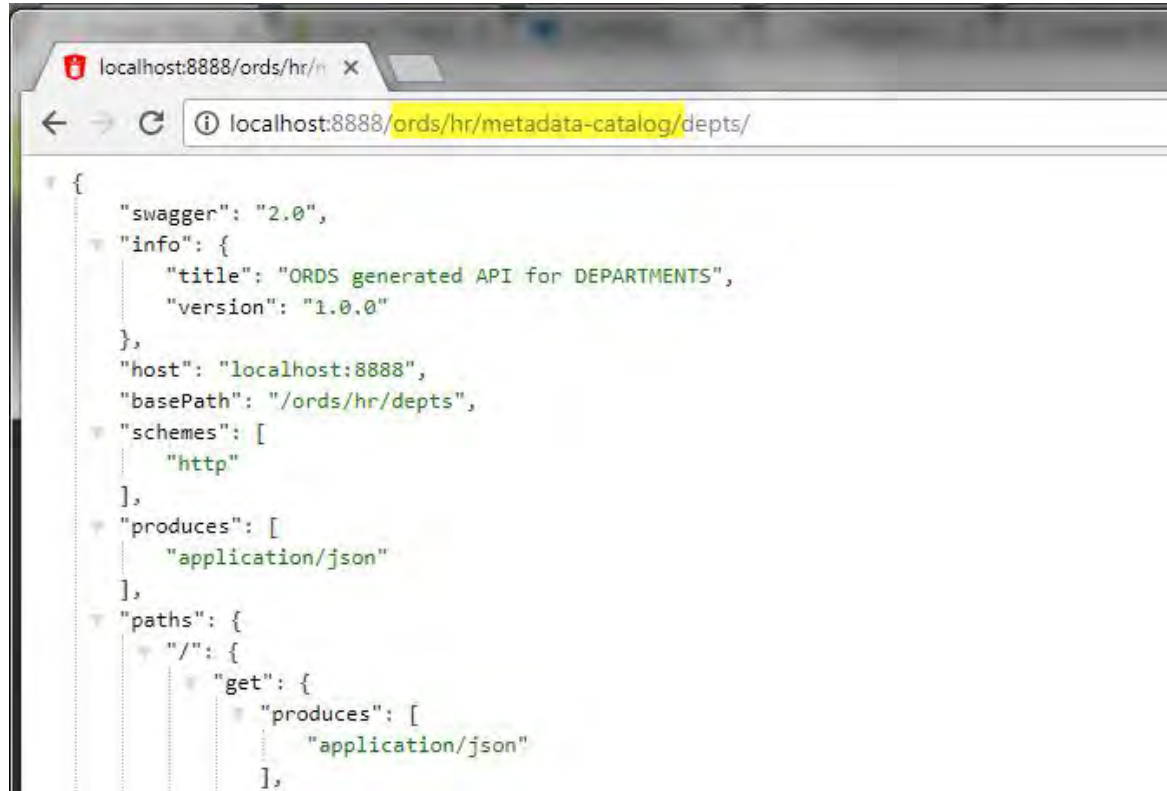
ID	TITLE	CONTENT_TYPE	IMAGE
1	ctrl-cl...	image/gif	(BLOB)

Red arrows indicate the flow of data from the REST client to the SQL Developer script and then to the table data.

Early Adopter (Beta) 17.3

- Swagger Support
- REST SQL
- REST Driver avail on OTN

Swagger/OpenAPI



The screenshot shows a web browser window with the address bar containing `localhost:8888/ords/hr/metadata-catalog/depts/`. The page content displays a JSON object representing the Swagger/OpenAPI specification for the ORDS generated API for DEPARTMENTS. The JSON is expanded to show its structure, including the Swagger version (2.0), API title and version, host, base path, supported schemes (http), and the path `/` with its `get` method details.

```
{
  "swagger": "2.0",
  "info": {
    "title": "ORDS generated API for DEPARTMENTS",
    "version": "1.0.0"
  },
  "host": "localhost:8888",
  "basePath": "/ords/hr/depts",
  "schemes": [
    "http"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/": {
      "get": {
        "produces": [
          "application/json"
        ]
      }
    }
  }
}
```

ORDS generated API for DEPARTMENTS 1.0.0

[Base URL: localhost:8888/ords/hr/depts]

Schemes

HTTP

default

GET /

POST /

GET /{id}

PUT /{id}

Parameters

Cancel

Name	Description
id * required	implicit
string (path)	120

ORACLE

PUT /{id}

Cancel

Parameters

Name	Description
id * required	implicit
string (path)	120
payload * required	Example Value (body)

```
{
  "DEPARTMENT_ID": 120,
  "DEPARTMENT_NAME": "social media",
  "MANAGER_ID": 101,
  "LOCATION_ID": 1700
}
```

Cancel

Parameter content type

application/json

Execute

Clear

Responses

Response content type

application/json

Curl

```
curl -X PUT "http://localhost:8888/ords/hr/depts/120" -H "accept: application/json" -H "content-type: application/json" -d '{"DEPARTMENT_ID": 120, "DEPARTMENT_NAME": "social media", "MANAGER_ID": 101, "LOCATION_ID": 1700}'
```

Server response

Code Details

200

Response body

```
{
  "department_id": 120,
  "department_name": "social media",
  "manager_id": 101,
  "location_id": 1700,
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8888/ords/hr/depts/120"
    }
  ]
}
```

REST...SQL???

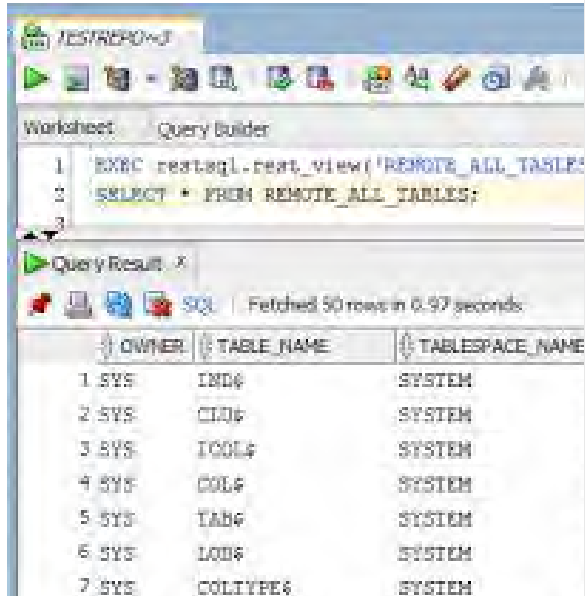
```
$ curl -X "POST" "http://localhost:9090/ords/hr/_/sql" \  
  -H "Content-Type: application/sql" \  
  -u HR:oracle \  
  -d '$select * from dual;'
```

- OFF by default
- DB Auth Support
- Ad-Hoc SQL/Anon Block/SQLcl Scripts
- stateless

REST SQL: Use Cases

- SQL Developer Web
- Access to Remote D

← Oracle Tool Tips @dermotoneill
<http://dermotoneill.blogspot.co.uk/2017/09/a-rest-enabled-sql-view.html>



The screenshot shows the SQL Developer interface. The 'Query Result' window displays the following data:

OWNER	TABLE_NAME	TABLESPACE_NAME
1 SYS	IND\$	SYSTEM
2 SYS	CLD\$	SYSTEM
3 SYS	ICDL\$	SYSTEM
4 SYS	COL\$	SYSTEM
5 SYS	TAB\$	SYSTEM
6 SYS	LOG\$	SYSTEM
7 SYS	COLTYPE\$	SYSTEM

```
48 p_username VARCHAR2,  
49 p_password VARCHAR2,  
50 p_sql CLOB)  
51 RETURN CLOB  
52 AS  
53  
54 v_columns CLOB := 'COLUMNS(';  
55 v_columnDetails VARCHAR2(4000);  
56 v_columnIsNull VARCHAR2(4000);  
57 v_columnPath VARCHAR2(4000);  
58 v_notFirstLoop BOOLEAN := FALSE;  
59  
60 BEGIN  
61 FOR metadata IN  
62 (SELECT columnName,  
63 columnTypeName,  
64 jsonColumnName,  
65 PRECISION,  
66 scale,  
67 isNullable  
68 FROM JSON_TABLE( request(p_url, --rest enabled sql url  
69 p_username, --username  
70 p_password, --password  
71 p_sql), --sql query  
72 '$.items.resultSet.metadata[*]') --path to resultset metadata  
73 COLUMNS(columnName VARCHAR2(4000) PATH '$.columnName', jsonColumnName VARCHAR2(4000) PATH '$.jsonColumnName'  
74 )  
75 LOOP  
76 v_columnDetails := '';
```

SQL Developer, On the Web

- Runs out of Oracle REST Data Services (ORDS)
- Cloud First
- All HTML5/JS
- Fully Functional SQL Worksheet
- Relational Diagrams for your database objects
- Real Time SQL Monitoring
- Alert Log Viewer, Storage Info, Basic DB Activity Overview

SQL Developer, In Your Browser

The screenshot displays the Oracle SQL Developer web interface. At the top, there is a navigation bar with tabs for 'Home', 'Worksheet', 'DBA', 'OS', and 'Data Modeler'. The 'DBA' tab is active, and a dropdown menu is open, listing various database management tasks such as 'Overview', 'Manage', 'Listener', 'Backups', 'Alerts', 'Sessions', 'Storage', 'Waits', 'Movements', 'Parameters', and 'Real Time SQL Monitoring'. The 'Overview' option is currently selected.

Below the navigation bar, the interface is divided into several panels:

- Recently Modified Objects:** A panel showing a tree view of database objects. It includes nodes for 'TEST_P' (dated 2017-09-25) and 'CREATE_T' (dated 2017-09-25). A timeline at the bottom shows dates from 9/24 to 9/27.
- Invalid Objects:** A panel titled 'Invalid Objects' with columns for 'Object', 'Type', and 'Status'. It currently displays 'No invalid objects' and a pagination control showing 'Page 1 (0 of 0 items)'.
- Table Stats Freshness:** A table showing the freshness of various tables. The table has columns for 'Table', 'Rows', 'Blocks', and 'Last Analyzed'.

Table	Rows	Blocks	Last Analyzed
Testing	2	5	< Week
T1	3	5	< Week
SQLDEV_JOB_FILES	0	0	< Month
SQLDEV_JOBS	0	0	< Month

SQL Developer SQL Worksheet Features

- Run SQL & PL/SQL
- Explain Plans & DBMS_OUTPUT
- Recover worksheets/code from prior sessions
- SQL History/Recall
- Code Insight
- Formatting and Code Parser/Error Highlighting
- Script & Query Output (text vs grid)
- Grid Exports
- Drag and Drop Code Generation AND MORE!

Navigator Worksheets

C##DBAAS_MONITOR

Tables

Search...

- ▶ ADHOC_TABLE_DATE
- ▶ ADHOC_TABLE_SIMPLE
- ▶ COUNTRIES
- ▶ DEPARTMENTS
- ▶ DEPARTMENTS_COPY
- ▶ EMPLOYEES
- ▶ EMPS_COPY
- ▶ FOO
- ▶ JOBS
- ▶ JOB_HISTORY
- ▶ KLR_LINKS
- ▶ KLR_MILESTONES
- ▶ KLR_PROJECTS
- ▶ KLR_TASKS
- ▶ KLR_TEAM_MEMBERS
- ▶ KLR_TO_DOS
- ▶ KRIS_TEST1

[Worksheet]*

```

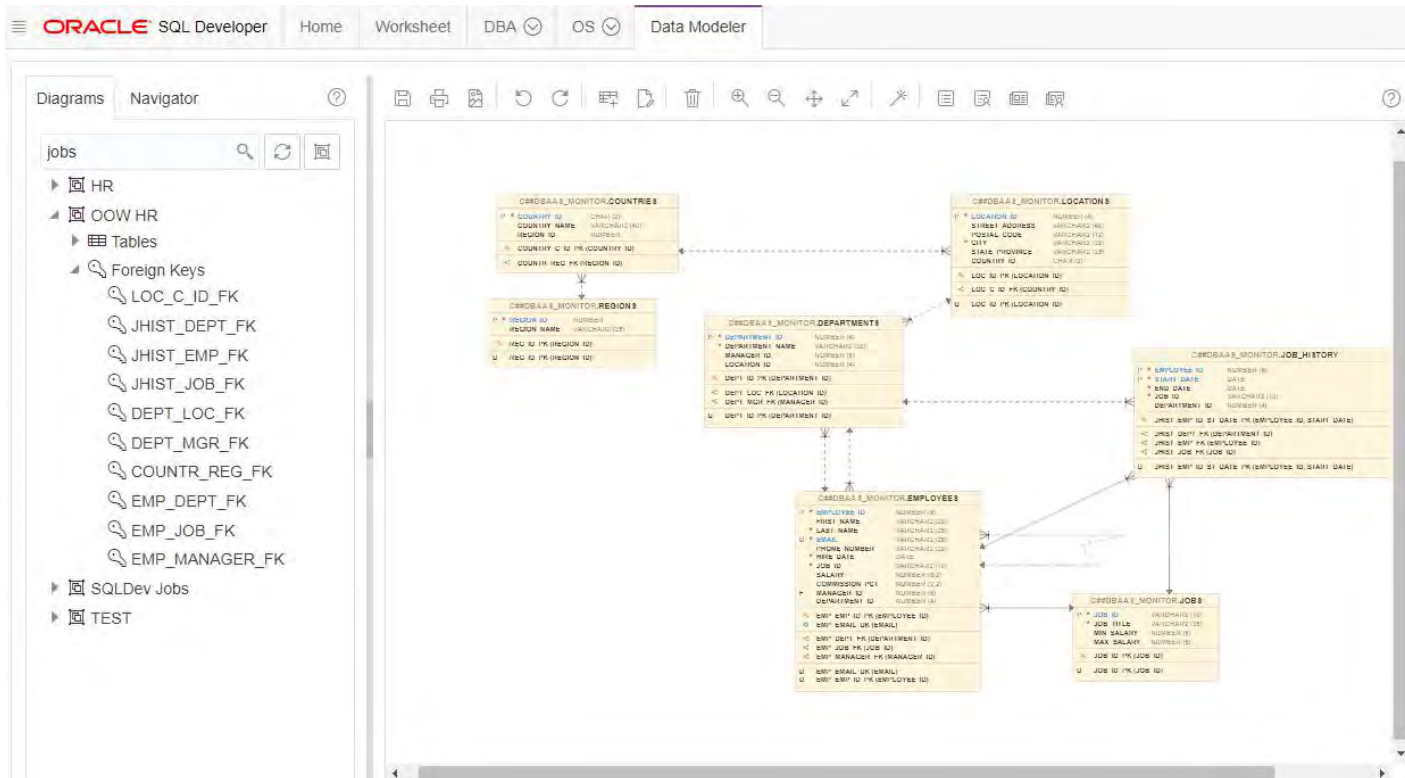
1
2 select
3
4     COMMISSION_PCT
5     DEPARTMENT_ID
6     EMAIL
7     EMPLOYEE_ID
8     FIRST_NAME
9     HIRE_DATE
10    JOB_ID
11    LAST_NAME
12    MANAGER_ID
13    PHONE_NUMBER
    
```

Query Result Script Output DBMS Output Explain Plan Autotrace SQL History



	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	100	Steven	King	SKING	515.123.4567	06/17/87 12:00 AM	AD_PRES
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	09/21/89 12:00 AM	AD_VP
3	102	Lex	De Haan	LDEHAAN	515.123.4569	01/13/93 12:00 AM	AD_VP
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	01/03/90 12:00 AM	IT_PROG
5	104	Bruce	Ernst	BERNST	590.423.4568	05/21/91 12:00 AM	IT_PROG
6	105	David	Austin	DAUSTIN	590.423.4569	06/25/97 12:00 AM	IT_PROG

Diagram Your Objects



SQL Developer Data Modeling v1 Features

- Create any combination of objects
- Create one or more diagrams
 - preview diagrams
 - search diagrams
- View all metadata/properties of object
- Generate DDL

ORACLE®