OpenWorld 2016
Querying Hadoop/HDFS from PL/SQL

CON6359

Nicholas Van Wyen
MTI
September 18, 2016

ORACLE
OPEN
WORLD

September 18–22, 2016
San Francisco

Accelerate Your
Digital Transformation
in the Cloud

Agenda

1 ▶ The Real-World

2 ▶ The Problem

3 ▶ The Solution

4 ▶ Considerations

5 ▶ Questions

Let's get started

1 The Real-World

2 The Problem

3 The Solution

4 Considerations

5 Questions

The Real-World

Different solutions, for different requirements
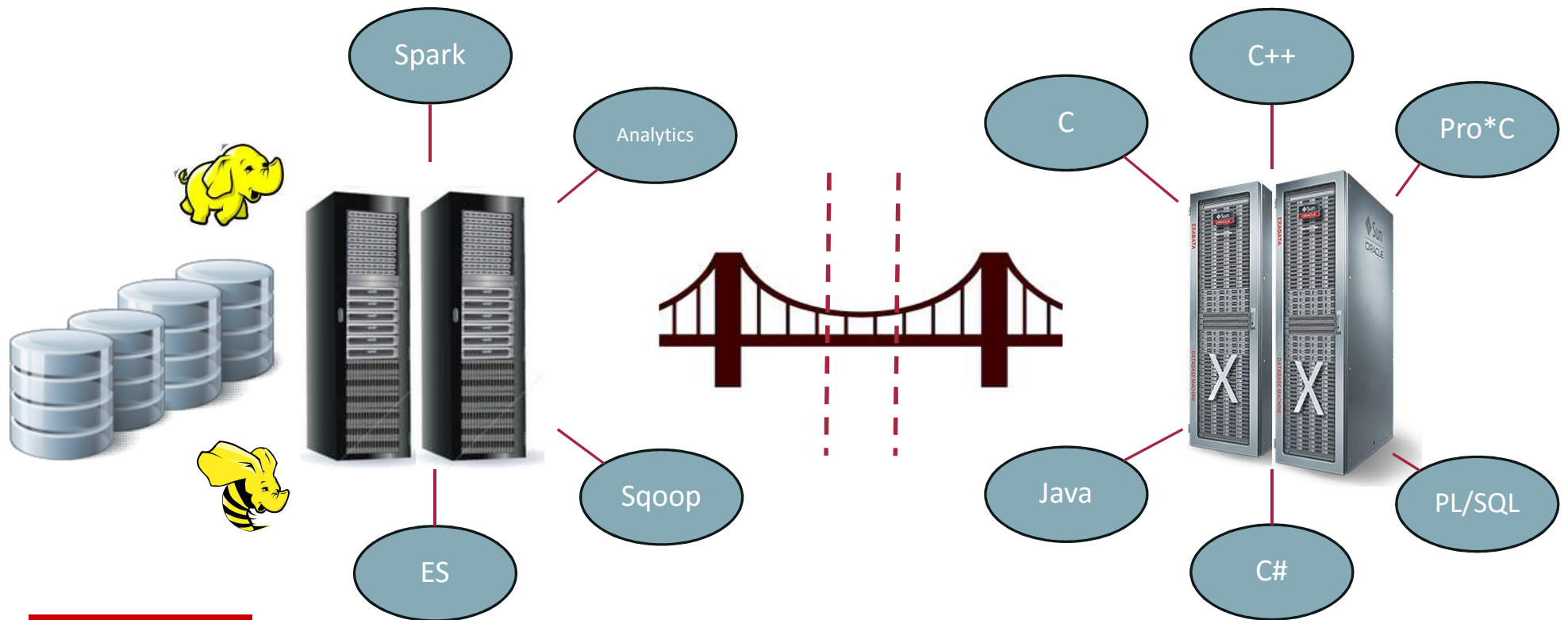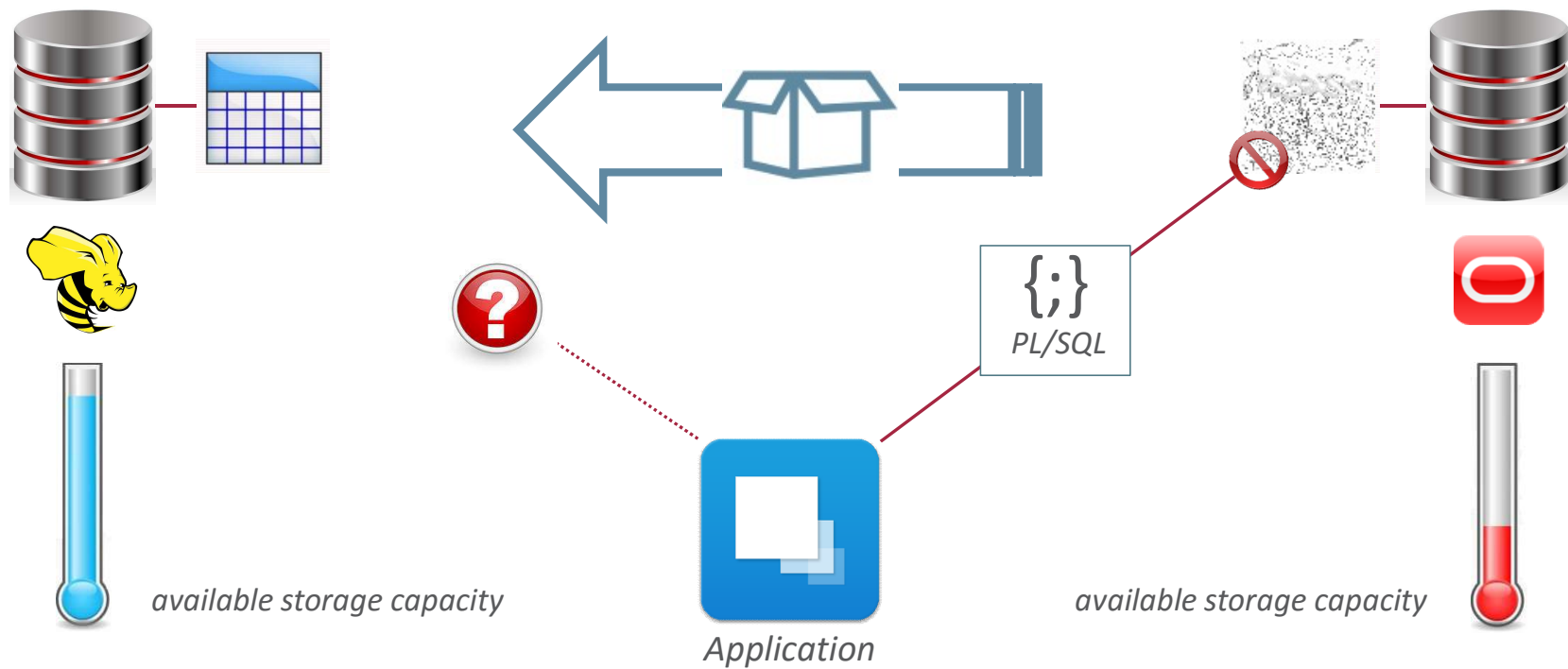
Moving on

1 The Real-World

2 The Problem

3 The Solution

4 Considerations

5 Questions

ORACLE®

The Problem

Changes

available storage capacity

{;}
PL/SQL

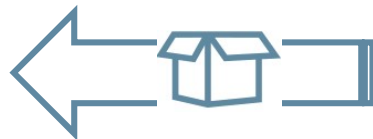*Application*

available storage capacity

The Problem

Example

```
$ beeline -u jdbc:hive2://hive.corp.com:10000 \
        -n oracle -w welcome1.passwd

 0: jdbc:hive2://localhost:10000> desc user_log;
+------------------+------------+---------+
|     col_name     | data_type  | comment |
+------------------+------------+---------+
| stamp            | date       |         |
| account          | string     |         |
| message          | string     |         |
+------------------+------------+---------+
```

```
SQL> desc SCOTT.USER_LOG

  Name           Null?      Type
  ----------     --------   --------------
  STAMP          NOT NULL   DATE
  ACCOUNT                   VARCHAR2(30)
  MESSAGE                   VARCHAR2(4000)
```

```
create view user_log_monthly
as
select stamp,
       account,
       message
  from scott.user_log
 where stamp between sysdate - 30
                 and sysdate;
```

```
procedure user_report( p out xmltype ) is
begin

    for rec in ( select account,
                        message
                   from scott.user_log
                  order by account ) loop
    ...

end user_report;
```
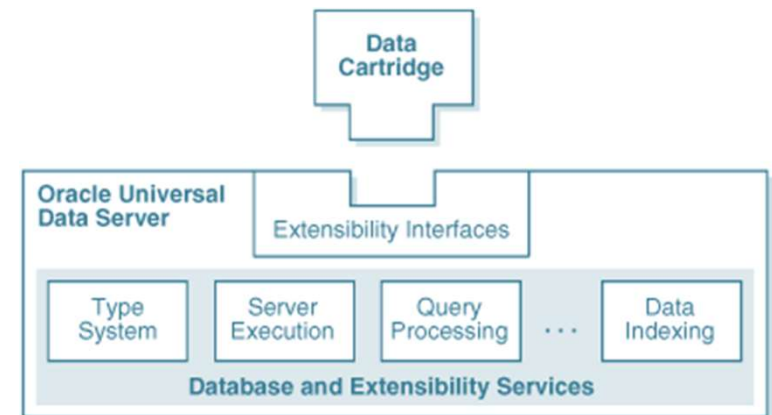
Next

1 The Real-World

2 The Problem

3 The Solution

4 Considerations

5 Questions

The Solution

Introduction

- Presenting Hive-ODCI
  - Built on Oracle Data Cartridge Interface
  - Dynamically accessing Hadoop/Hive within an Oracle 12c RDBMS
    - A JDBC Hadoop/Hive connector
  - Allow for First-Class Oracle objects
  - Leveraging existing RBAC
  - Support for active Bind variables
    - User defined, Static or Saved
  - Native support for SQL and PL/SQL
  - Easy to use and integrate

The Solution

Overview

The Solution

Example

bidirectional

pipelined data

hive-odci

hive_t

hive_q

session

binding

param

Application

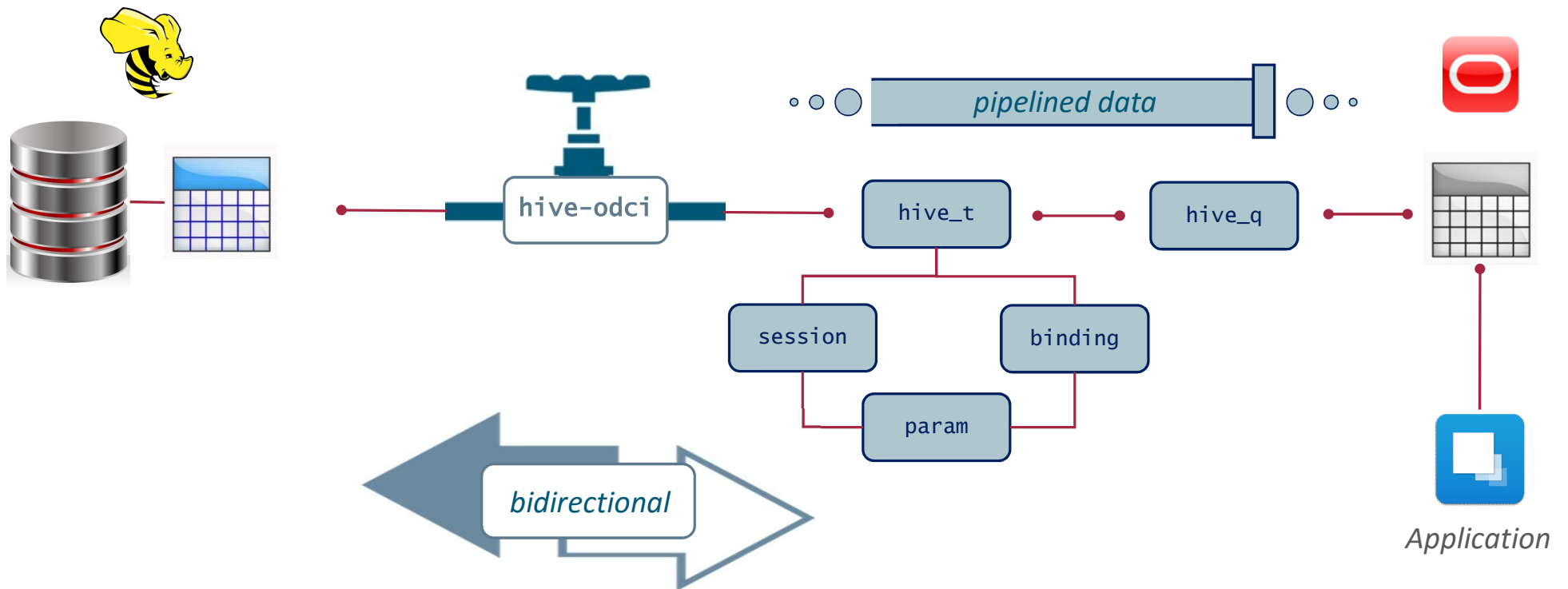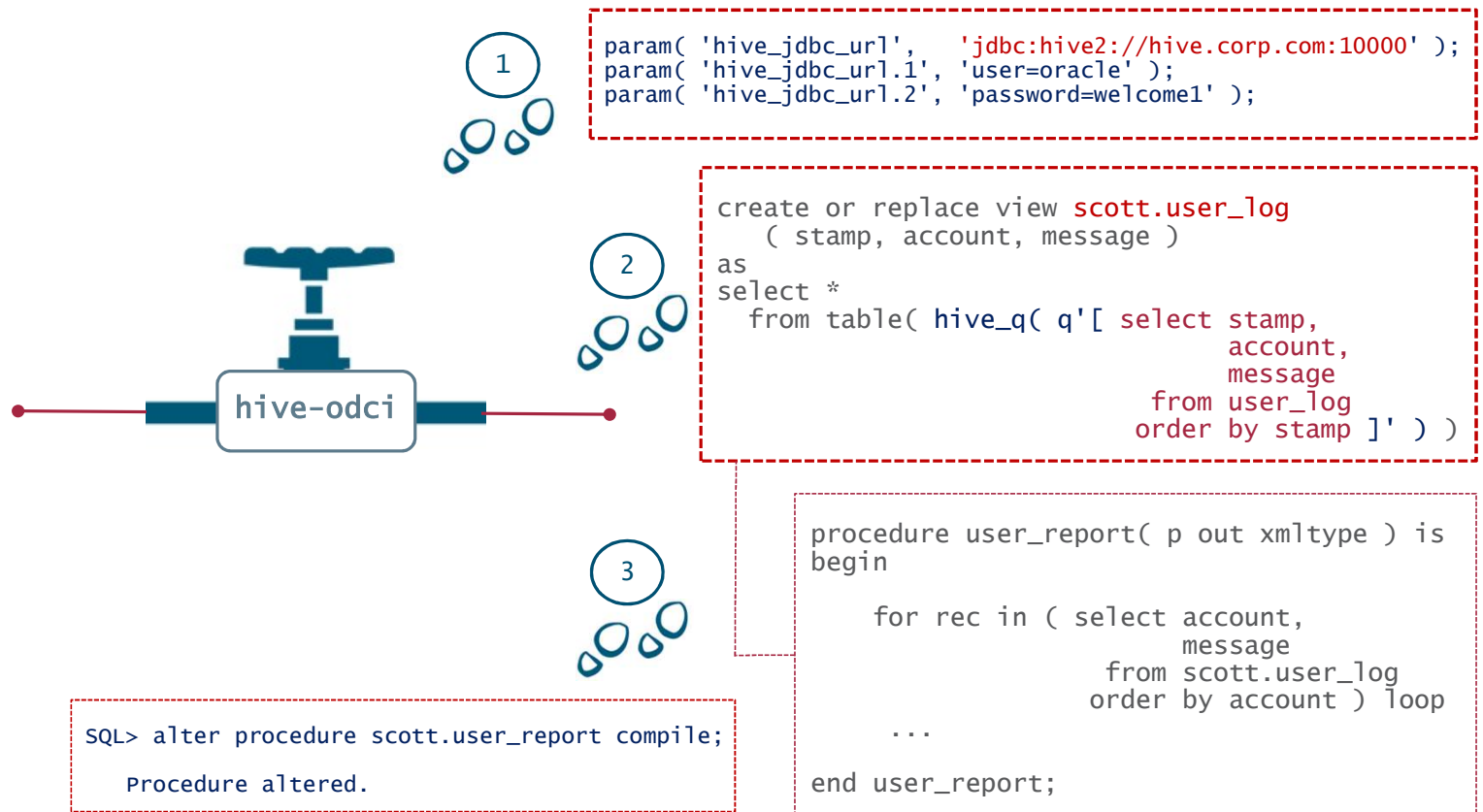# The Solution

Example

1

```
param( 'hive_jdbc_url',   'jdbc:hive2://hive.corp.com:10000' );
param( 'hive_jdbc_url.1', 'user=oracle' );
param( 'hive_jdbc_url.2', 'password=welcome1' );
```

2

```
create or replace view scott.user_log
    ( stamp, account, message )
as
select *
    from table( hive_q( q'[ select stamp,
                                   account,
                                   message
                              from user_log
                          order by stamp ]' ) )
```

hive-odci

3

```
procedure user_report( p out xmltype ) is
begin

    for rec in ( select account,
                        message
                   from scott.user_log
               order by account ) loop

    ...

end user_report;
```

```
SQL> alter procedure scott.user_report compile;

    Procedure altered.
```
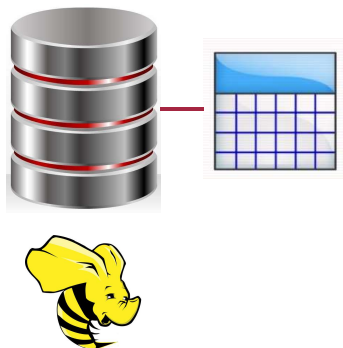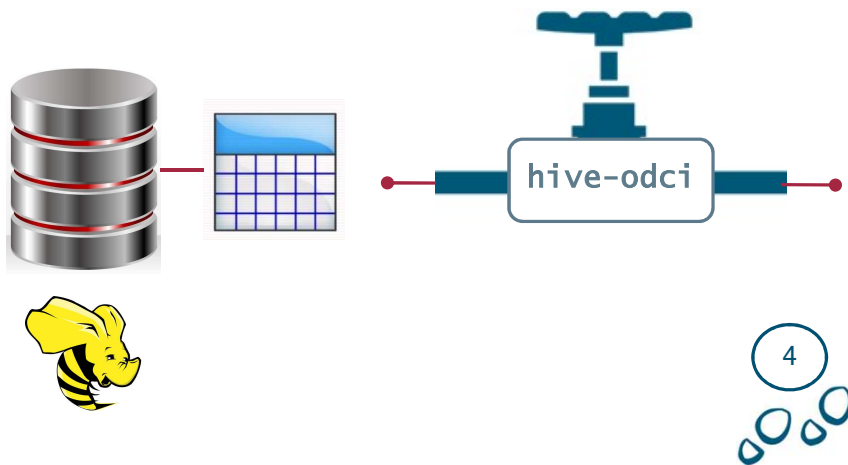
The Solution

Example

```
create or replace view scott.user_log_monthly
        (
            stamp,
            account,
            message
        )
    as
    select *
      from table( hive_q( q'[ select stamp,
                                      account,
                                      message
                                 from user_log
                                where stamp between ? and ? ]',
                  hive_binds( hive_bind( to_char( sysdate - 30,
                                                  'yyyy-mm-dd' ),
                                         1 /* type_date */,
                                         1 /* ref_in */ ),
                              hive_bind( to_char( sysdate, ,
                                                  'yyyy-mm-dd' ),
                                         1 /* type_date */,
                                         1 /* ref_in */ ) ) )
```
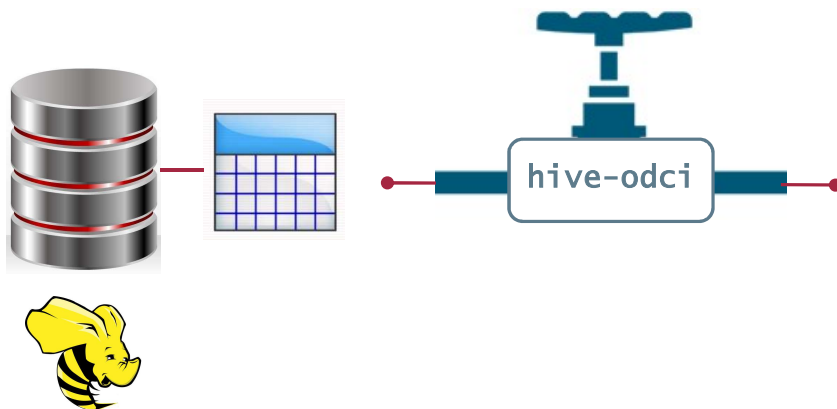
hive-odci

4

The Solution

Example


```
create or replace trigger scott.user_log_dml
    instead of insert or update or delete on scott.user_log
    for each row
declare
    cmd varchar2( 4000 );
    bnd hive_binds := hive_binds();
begin

    if ( inserting ) then

        cmd := q'[ insert into user_log
                        ( stamp, account, message )
                    values
                        ( ?, ?, ? ) ]';

        bnd.extend;
        bnd( bnd.count ) := hive_bind( to_char( :new.stamp,
                                                'yyyy-mm-dd' ),
                                        hive_binding.type_date,
                                        hive_binding.ref_in );

        ...
    elsif ( updating ) then
        ...
    end if;

    hive_remote.dml( cmd, bnd );

end user_log_dml;
```

Wrapping it up

1 ▶ The Real-World

2 ▶ The Problem

3 ▶ The Solution

4 ▶ Considerations

5 ▶ Questions

ORACLE®

Considerations

In Oracle

- Become familiar with the Hive-ODCI API
  - Read the documentation
  - Ask questions and test, test, test

- Use session isolation whenever possible
  - Particularly authentication, set at the session not the system

- Lean on your experience and your DBA Team
  - Keep signatures consistent
  - Performance tune
  - Change code if necessary

Considerations

In Hive

- Analytics over in-line views
  - Review queries and use common sense
- Leverage the CBO and gather statistics
- Use best practices
  - ORCFile
  - Apache Tez
  - Vectorized queries
- Lean on your experience and your BDS Team

Considerations

Reach out

- If you have questions, concerns or comments
  - Feel free to contact me

- Available on Github
  - https://github.com/nvanwyen/hive-odci
  - https://github.com/nvanwyen/hive-odci/releases/latest

- Contact
  - nvanwyen@mtihq.com

That's it

1 ▸ The Real-World

2 ▸ The Problem

3 ▸ The Solution

4 ▸ Considerations

5 ▸ Questions