NCOAUG

NORTH CENTRAL ORACLE APPS USER GROUP

TRAINING DAY

AUGUST 1, 2019

Achieving Massive Scalability And Total Fault Isolation Through Oracle Sharding

Kai Yu

Distinguished Engineer

Database *Solutions Engineering, Dell EMC*

Kai Yu

ORACLE ACE Director

DELL EMC

- Distinguished Engineer, Dell EMC Database Engineering
- 25+ years working in IT Industry
- Specializing in Oracle Database, Cloud, Virtualization
- Author and Speaker at IEEE and Oracle Conferences
- IOUG Cloud Computing SIG Co-founder and VP
- Oracle ACE Director
- Co-recipient of the 2011 OAUG Innovator of Year
- 2012 Oracle Excellence Award- Technologist of the
- Year: Cloud Architect by Oracle Magazine
- My Blog: http://kyuoracleblog.wordpress.com/

IEEE TRANSACTIONS ON BIG DATA

ORACLE OPEN WORLD

IEEE HPCC

UK OUG

SELECT

SELECT

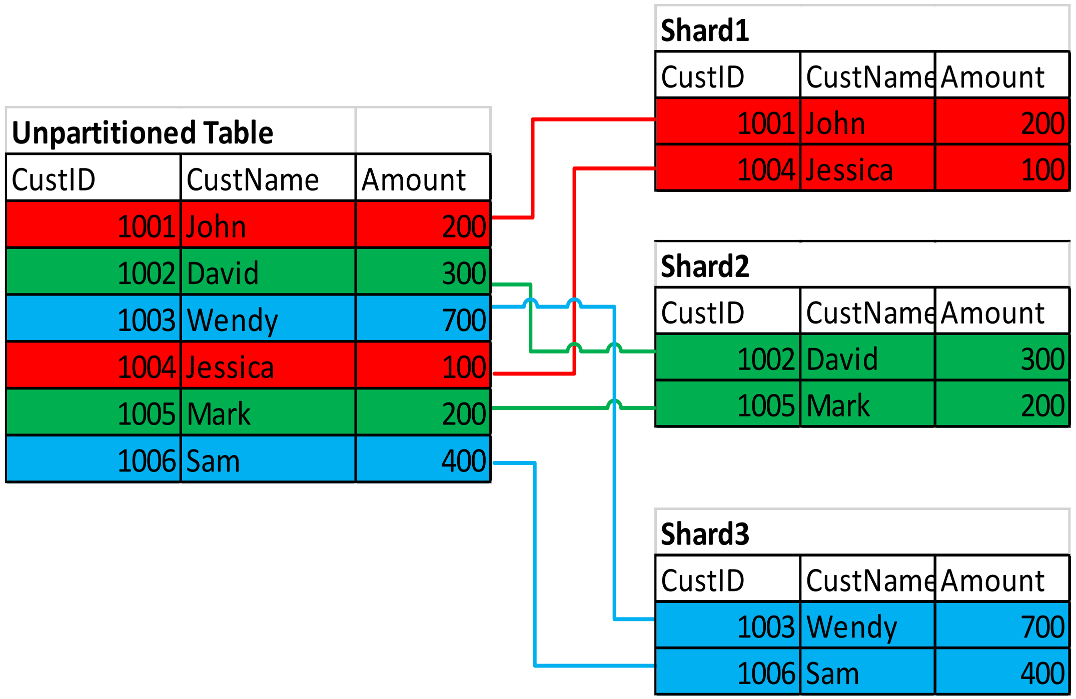
Agenda

- Oracle Sharding Architecture Overview
- Deployment of Oracle Sharded Databases
- An Oracle Sharding POC project

Oracle Sharding Architecture Overview

- A data tier architecture introduced in 12.2 , data horizontally partitioned across independent databases.
- Data horizontally partitioned
  - Table split : sharded table
  - Same columns, subset of rows
- Multiple sharded databases
  - Each in dedicated server
  - Own local hardware
  - Shared nothing between sharded databases
  - Different architecture from RAC
  - Compare with Table Partitioning
- Application point of view:
  - Logical group as a single database
  - Data partition transparent to application

Horizon Table Partition in Sharding



Unpartitioned Table

CustID	CustName	Amount
1001	John	200
1002	David	300
1003	Wendy	700
1004	Jessica	100
1005	Mark	200
1006	Sam	400

Shard1

CustID	CustName	Amount
1001	John	200
1004	Jessica	100

Shard2

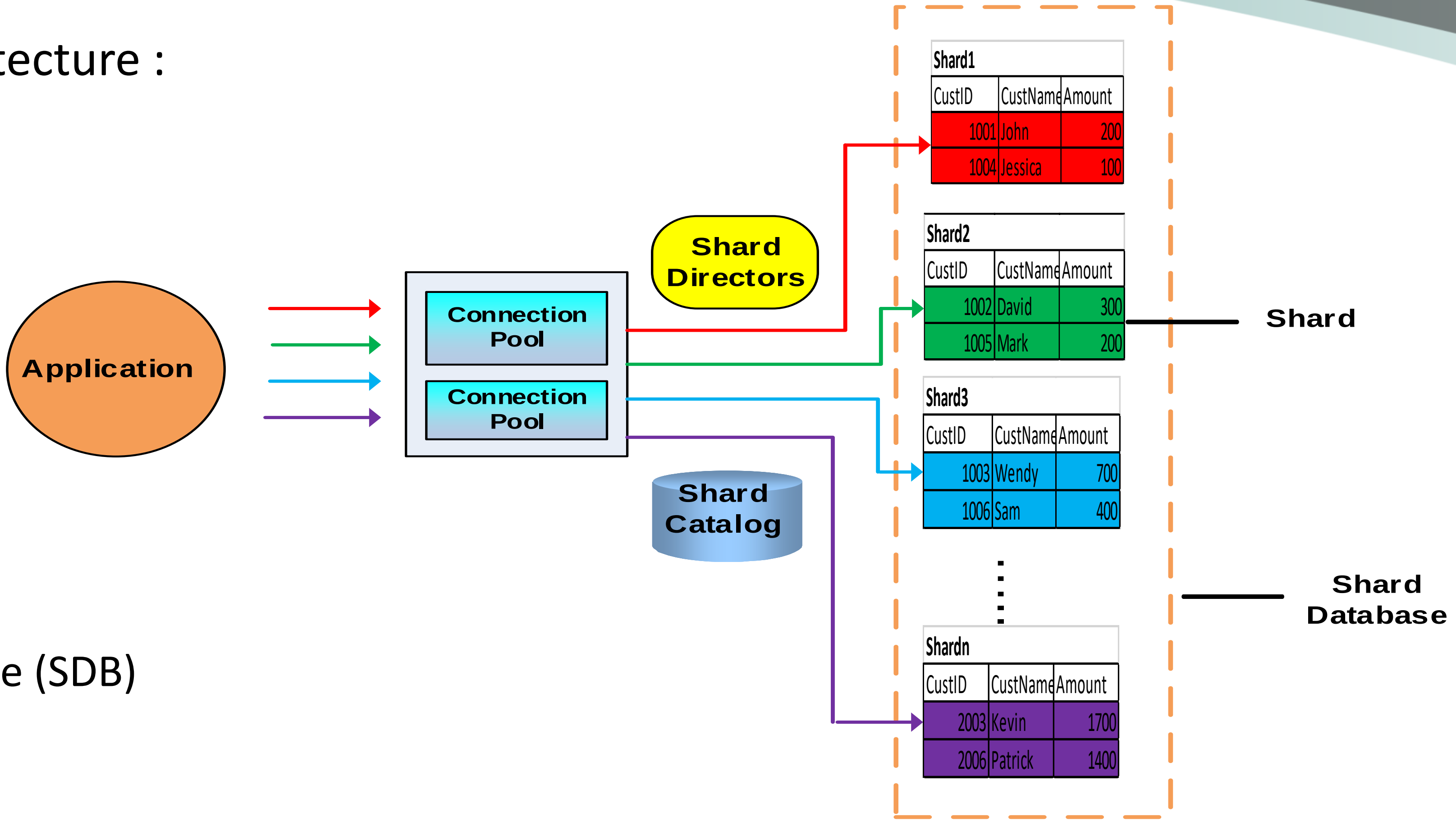
CustID	CustName	Amount
1002	David	300
1005	Mark	200

Shard3

CustID	CustName	Amount
1003	Wendy	700
1006	Sam	400

Oracle Sharding Architecture Overview

Oracle Sharding Architecture :



Shard1

CustID	CustName	Amount
1001	John	200
1004	Jessica	100

Shard2

CustID	CustName	Amount
1002	David	300
1005	Mark	200

Shard3

CustID	CustName	Amount
1003	Wendy	700
1006	Sam	400

Shardn

CustID	CustName	Amount
2003	Kevin	1700
2006	Patrick	1400

- Components:
  - Sharding Database (SDB)
  - Shards
  - Shard Catalog:
  - Shard Director

Oracle Sharding Architecture Overview

Oracle Sharding: major components:

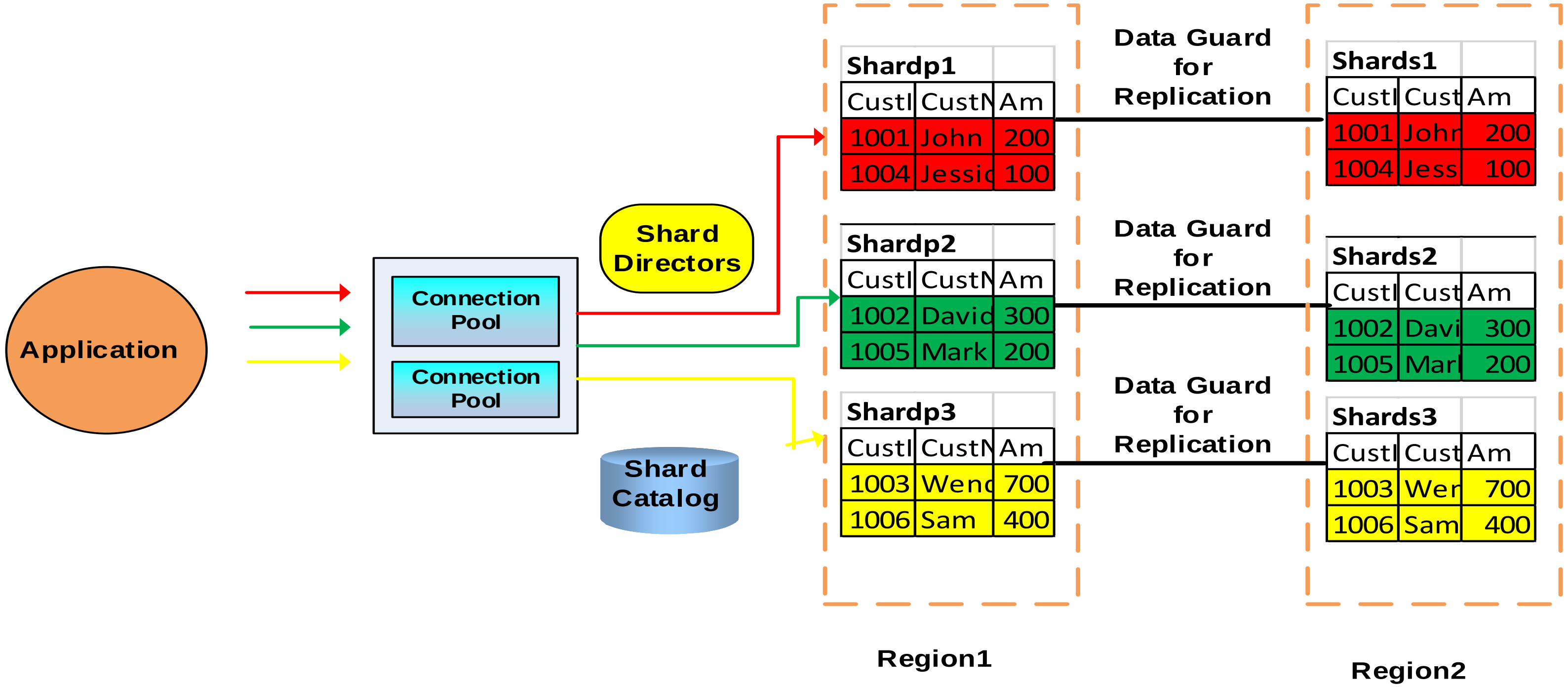
- Sharding Database (SDB): a single logical database horizontally partitioned across of pool of Oracle Databases(Shards)
- Shards
  - Independent physical Oracle database, Host a subset of the sharded database
  - With local resources: CPU, memory, disks (no shared disks)
  - Shared no hardware and software between shards
  - Shards are replicated for HA or DR with Oracle replication.
  - Standby/primary: same region for HA; different regions for DR
- Global Service Manager (GSM)/Shard Director
  - **GSM**: route connections based on database roles, load, replication lag locality. Transfer the connections based on the real-time load balancing
  - **Shard Director**: a specific implementation of GSM for sharding
    - Maintain the current topology map of SDB
    - Route connections to appropriate shard based on the sharding key
    - Deploy multiple Shard Directors for HA purpose.

Oracle Sharding Architecture Overview

- Shard Director (continued)
  - Maintain the SDB configuration and availability of shards in runtime
  - Monitor network latency between its regions
  - Regional listener for clients to connect to an SDB
  - Manage global services
  - Perform connection load balancing
- Shard Catalog:
  - A special purpose Oracle Database required by GSM
  - Stores the configuration of GSM, database and service metadata
  - where all configuration shards/global service change are initiated
  - All DDLs in a SDB executed by connecting to the shard Catalog
  - Stores the master copy of all duplicated tables in SDB
  - Automatically replicate changes to duplicated tables in all shards
  - Automated shared deployment, centralized management,
- Management of SDB: EM Cloud control and GDSCTL (command-line interface)

Oracle Sharding Architecture Overview

- Oracle Sharding integrated with replication for High Availability or DR



Shardp1		
CustI	CustN	Am
1001	John	200
1004	Jessic	100

Shardp2		
CustI	CustN	Am
1002	David	300
1005	Mark	200

Shardp3		
CustI	CustN	Am
1003	Wend	700
1006	Sam	400

Shards1		
CustI	Cust	Am
1001	John	200
1004	Jess	100

Shards2		
CustI	Cust	Am
1002	Davi	300
1005	Mar	200

Shards3		
CustI	Cust	Am
1003	Wer	700
1006	Sam	400

- Supported Replication
  - Oracle Data Guard
  - Oracle GoldGate
- Shardgroups: primary shardgroupP has: shardp1, shardp2
  secondary shardgroupS has: shards1, shards2
- Primary and Secondary can be same/region(data center) or different regions(data centers)

Applications Suitable for Oracle Sharding

- Benefits of the Oracle Sharding
  - Linear Scalability: make it possible to linearly scale the performance isolating availability and performance issue in a shard
  - Fault isolation

    Sharing nothing architecture between shard isolates the availability and performance issue and maintenance within a shard.
  - Support HA and DR with the replication between shards.
  - Data dependent routing automatically routines connections to proper shards
  - Geographical distribution
  - Simplify deployment and life cycle management
- Applications suitable for Oracle Sharding
  - OLTP applications that are suitable for sharding
  - Well-defined Data model and data distribution: eg sharding key

    Data associated with a single value of sharding key such as online store application: customer_id as sharding key

.

Shard Database Schema Design

- Sharded tables
  - A shard table is partitioned into smaller pieces in multiple shards
  - SQL issued by applications don’t have to refer to shard
  - Partitions distributed across shards at the tablespace level by shard key: each partition is in one tablespace unit in one shard:
  - Example: Create SHARD Table customers

```
          (custID    Number NOT Null,
           Custname   VARCHAR2(40),
          Amount    Number
          CONSTRAINT cust_pk PRIMARY KEY(custID))
          PARTITION BY CONSISTENT HASH (custID)
          PARTITIONS AUTO TABLESPACE SET tsp1;
```

  - Sharded table family: a set of tables that are shared in the same way:

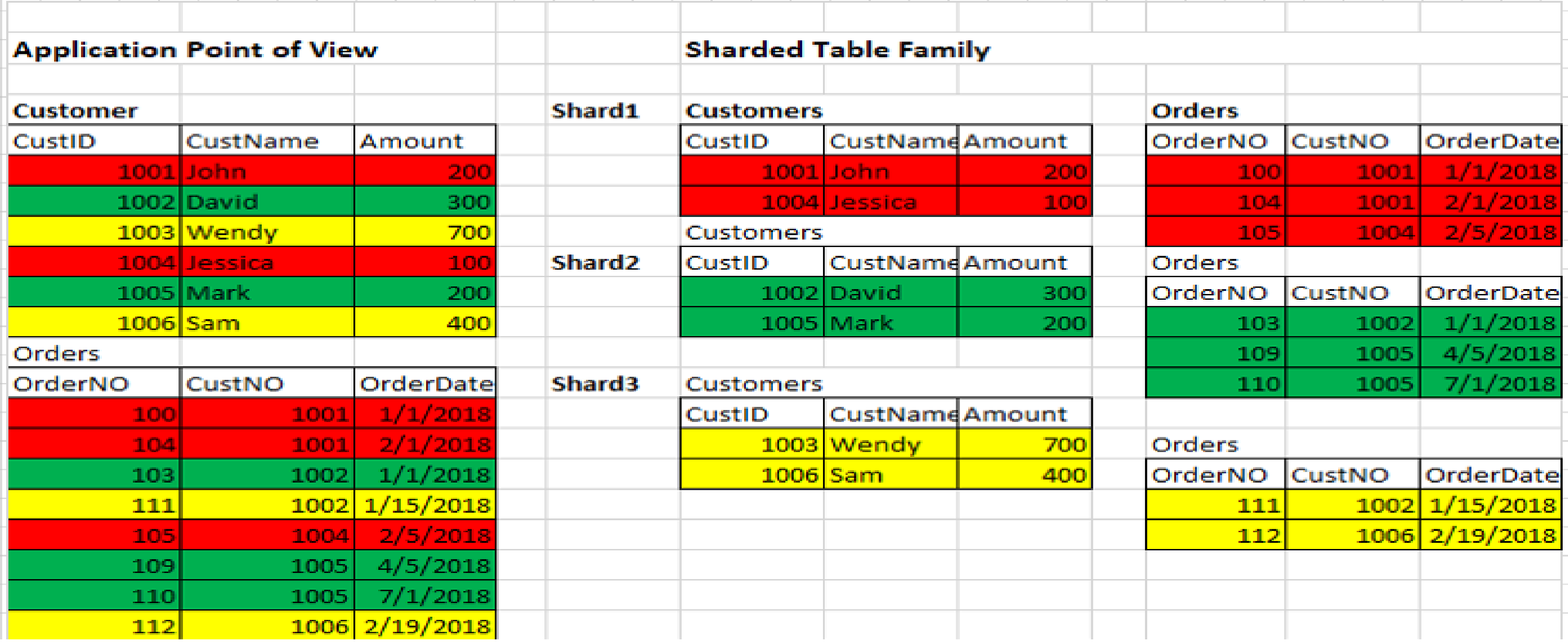
Example: customer table sharded with sharding key custID

its child tables: orders(orderNO, custID, orderDate)

order_items(itemNO, orderNO, custID,prodID, quantity)

custID: sharding key for all three tables,

Shard Database Schema Design



Application Point of View

Customer

CustID	CustName	Amount
1001	John	200
1002	David	300
1003	Wendy	700
1004	Jessica	100
1005	Mark	200
1006	Sam	400

Orders

OrderNO	CustNO	OrderDate
100	1001	1/1/2018
104	1001	2/1/2018
103	1002	1/1/2018
111	1002	1/15/2018
105	1004	2/5/2018
109	1005	4/5/2018
110	1005	7/1/2018
112	1006	2/19/2018

Sharded Table Family

Shard1

Customers

CustID	CustName	Amount
1001	John	200
1004	Jessica	100

Orders

OrderNO	CustNO	OrderDate
100	1001	1/1/2018
104	1001	2/1/2018
105	1004	2/5/2018

Shard2

Customers

CustID	CustName	Amount
1002	David	300
1005	Mark	200

Orders

OrderNO	CustNO	OrderDate
103	1002	1/1/2018
109	1005	4/5/2018
110	1005	7/1/2018

Shard3

Customers

CustID	CustName	Amount
1003	Wendy	700
1006	Sam	400

Orders

OrderNO	CustNO	OrderDate
111	1002	1/15/2018
112	1006	2/19/2018

- A Primary key is sharding key or is prefixed by sharding key

customer table: custID: for primary key and sharding key

orders table: sharding key: custID, primary key: custID+orderNO

order_items tables: sharding key: custID

primary key: custID+orderNo +itemNo

Shard Database Schema Design

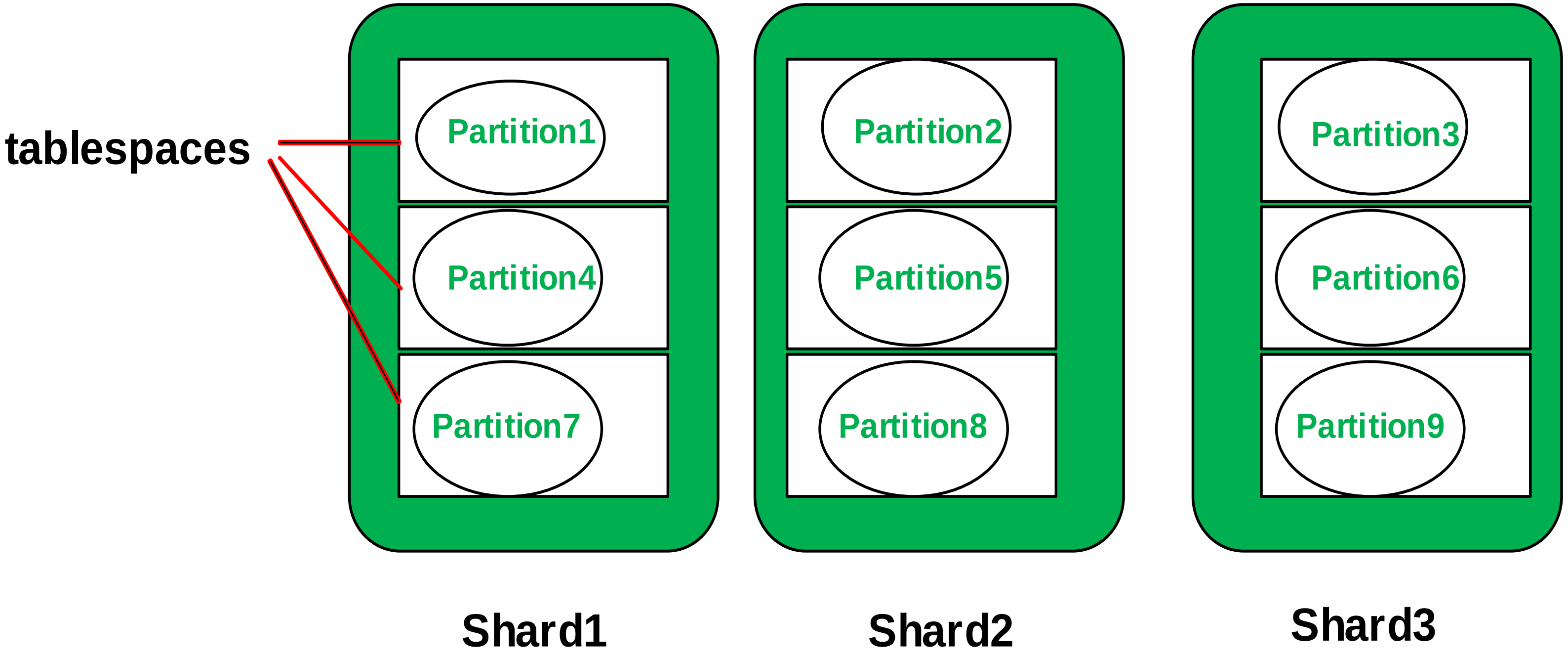
- Duplicated table
  - Tables have same contents and duplicated on all shards, such as reference information like Products table, shared by all customers in shards can not be sharded.
  - Synchronize the duplicated tables using materialized view
  - Create duplicated table:

    ```
    CREATED DUPLICATED TABLE products
    (ProdID Number PRIMARY KEY
     ProdName VARCHAR2(20), UnitPrice   Number(6,20))
    ```
- Non-table objects created on all shards:
  - non-table objects can be created in shards
  - Only table creation need extra key words: Sharded/Duplicated
  - Enable shard DDL for the session: alter session enable shard ddl;
- Execute DDL in a sharded database:
  - Use GDSTRL sql: GDSTL>*sql “create tablespace tspset”*
  - Connect to shardcatlog: *using sqlplus: SQL> create tablespace tspset;*

.

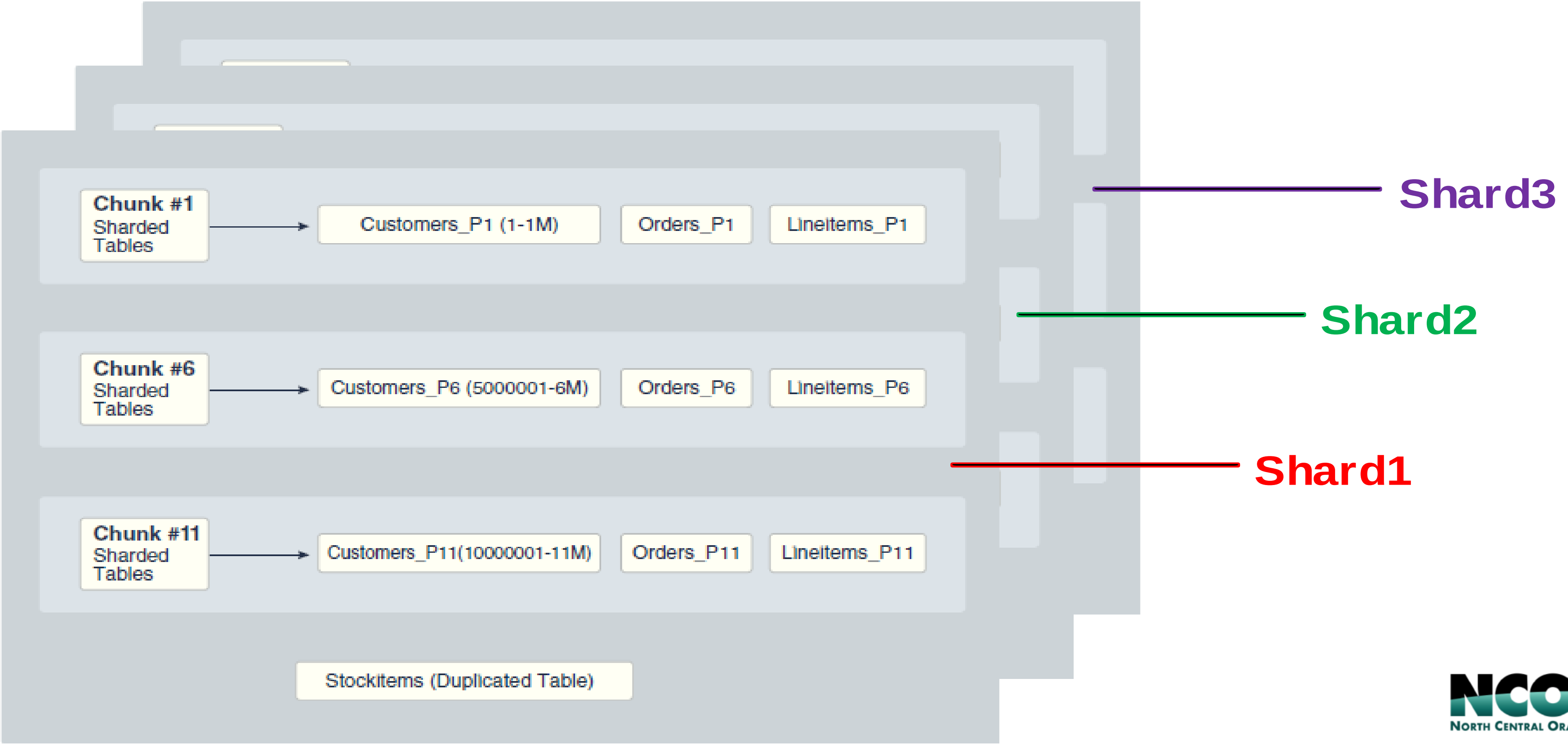
Shard Database: Physical Organization

- Partitions, tablespaces, shards
  - Sharding is distributed partitioning, different from table partition
  - Each partition is stored in a separate tablespace in shard.
  - Partitions are stored in a tablespace set that reside on different shards



Shard Database: Physical Organization

- Partitions, tablespaces , chunks, shards
  - Grouping corresponding partitions of a table family: a chunk
  - Contains a single partition of each table of a table family.



Sharding Methods

- System-managed sharding
    - Data is automatically partitioned and distributed across shards
    - Automatically maintain balanced distribution when adding/removing shards
    - Benefit: No need for users to involve data partitioning
    - Use consistent hash to map the partition key to # of partition
    - Use key word
      Partition by CONSISTENT HASH(SHARDKING_KEY)
- User-Defined Sharding
    - Users explicitly specify the mapping of data to individual shards
    - Users need to monitor and maintain the balanced distributions
    - Partitioned by range or list.

```
Create sharded table customers
 (custID          NUNBER,      Customer_name VARCHAR(30)
  state_code      VARCHAR(2),  Status             VARCHAR(1)
  Partition by List(state_code) )
```

Sharding Methods

```
(Partition p_pacific(‘CA’, ‘OR’,’WA’) tablespace ts_1
 Partition p_west(’UT’,’WY’,’CO’) tablespace ts_2
 Partition p_southwest(‘AZ’,’NM’,’TX’,) tablespace ts_3
 Partition p_east(‘NY’,’VM’,’NJ’,’MA’) tablespace ts_4)
```

- Use shardspace for user-defined sharding
  - a set of shards that stores the data of range or list of key values

It must be created and populated with shards in GDSCTL:

```
 ADD SHARDSPACE –SHARDSPACE pacific, west, east;
 ADD SHARD –CONNECT shard-1 –SHARDSPACE pacific;
ADD SHARD –CONNECT shard-2 –SHARDSPACE west;
ADD SHARD –CONNECT shard-3 –SHARDSPACE east;
```

Then create tablespace in shardspace:

```
CREATE TABLESPACE ts1 IN SHARDSPACE pacific;
CREATE TABLESPACE ts2 IN SHARDSPACE west;
CREATE TABLESPACE ts3 IN SHARDSPACE west;
CREATE TABLESPACE ts4 IN SHARDSPACE east;
```

Sharding Methods

Key values	Partition	Tablespace	Shardspace	Shard
(‘CA’, ‘OR’,’WA’)	p_pacific	*ts1*	pacific	shard-1
(’UT’,’WY’,’CO’)	p_west	ts2	west	shard-2
(‘AZ’,’NW’,’TX’)	p_southwest	ts3	west	shard-2
(‘NY’,’VM’,’NJ’,’MA’)	p_east	ts4	east	shard-3

- Composite sharding
  - Combination of user-defined and system-managed sharding
  - Take benefits of both sharding methods
  - Two levels of sharding:
    - Data first partitioned by list or range across multiple shardspaces
    - Then further partitioned by consistent hash across multiple shards in each shardspace.

Oracle 18c new features for Sharding:

- User-defined sharding
  - 12cR2 introduced system-managed sharding and composite sharding.
  - 18c introduced User-defined sharding
- PDB sharding
  - Allow to use PDB as shards or catalog database
  - 18c supports shard as a single PDB in a given CDB
  - A CDB can contain other non-shard PDBs
- RAC sharding
  - Enable to logically affinitize table partitions to RAC instance
  - Allows the requests that specify a sharding key are rounted to the instance
  - Enable RAC sharing feature: alter system enable affinity <table>
- Midtier sharding:
  - For multitier application, applications tier are sharded
  - *Topology server has sharded DB info, REST_API call to the Topology server to rount the request to the midtier associated with the sharding key*

Oracle 19c new features for Sharding:

- Multiple Table Family support for System-managed Sharding
  - 18c only one table family for each sharded database.
  - 19c supports for multiple table families, system-managed shared database only
- Support for multiple PDB-shards in the same CDB
  - 18c: a single PDB in a CDB can be used as shard or shard catalog databaase
  - 19c :more than one PDB in a CDB can be used as shard or shard catalog database

    a CDB contains shard PDBs from different sharded databases.
  - A CDB can contain other non-shard PDBs
- Unique Sequence numbers across Shards
  - 19c allow you to independently generate sequence number on each shard
  - Add shard and noshard clauses to sequence object DDL
- Enable multiple shards query coordinators on Oracle Active Data Guard standbys of the shard catalog database.
- Propagation of parameter settings across shards:

  allow DBSs to centrally mange and parameters settings: shard catalogs and shads

Oracle Sharding Deployment

- Options:
  - Deploy Oracle Sharding in Oracle Cloud: System Managed Sharding with Active Data Guard on Oracle Cloud (DBCS) Cookbook, Oracle Whitepaper
  - Try it in your own systems:
- Environment Settings
  - Software installations
  - Prerequisite checks and settings
  - Create sharding database
  - Create sharded tables

Oracle Sharding Deployment

- Environment Settings
  - One Shard Director (GSM) on fldcnode9
  - One Shard Catalog Database (SDB) on fldcnode9
  - Three Shards on fldcnode10, fldcnode11, fldcnode12
- Software Installations
  - Download and Install Oracle database software on all the hosts for both shard catalog and database shards
  - Download and install Oracle Global Service Manager on the hosts for shard directors

Oracle Sharding Deployment

- Prerequisite Checks and Settings
  - Disable iptable on Catalog and all Shard Databases

```
[root@fldcnode9]# service iptables stop
[root@fldcnode9]# service iptables status
iptables: Firewall is not running.
[root@fldcnode9]# chkconfig iptables off
```

  - Create necessary directories

```
[oracle@fldcnode9]$ export ORACLE_BASE=/u01/app/oracle
[oracle@fldcnode9]$ mkdir -p $ORACLE_BASE/oradata
[oracle@fldcnode9]$ mkdir -p $ORACLE_BASE/fast_recovery_area
```

Oracle Sharding Deployment

- Prerequisite Checks and Settings
  - Check ownership and permissions for following directories and files

```
chown root $ORACLE_HOME/bin/extjob
chmod 4750 $ORACLE_HOME/bin/extjob
chown root $ORACLE_HOME/rdbms/admin/externaljob.ora
chmod 640 $ORACLE_HOME/rdbms/admin/externaljob.ora
chown root $ORACLE_HOME/bin/jssu
chmod 4750 $ORACLE_HOME/bin/jssu

[oracle@fldcnode9]$ ls -ltr $ORACLE_HOME/bin/extjob
-rwsr-x--- 1 root oinstall 2592299 Sep 13 19:11 /u01/app/oracle/product/12.2.0/db_1/bin/extjob
[oracle@fldcnode9]$ ls -ltr  $ORACLE_HOME/rdbms/admin/externaljob.ora
-rw-r----- 1 root oinstall 1534 Dec 21  2005 /u01/app/oracle/product/12.2.0/db_1/rdbms/admin/externaljob.ora
[oracle@fldcnode9]$ ls -ltr $ORACLE_HOME/bin/jssu
-rwsr-x--- 1 root oinstall 2377646 Sep 13 19:11 /u01/app/oracle/product/12.2.0/db_1/bin/jssu
```

80cv

Oracle Sharding Deployment

- Create Shard database
  - Prepare Catalog database
  - Add GSM and Start GSM
  - Create Shard Catalog (SDB)
  - Create Shards
  - Deploy
  - Create and Start Global Services
  - Create Application User/Schema with Necessary Privileges
  - Create Tablespace for Shards
  - Create Sharded Tables and Duplicated Tables

Oracle Sharding Deployment

- Implementation
  - Prepare Catalog database
    - Create the database with DBCA to host the shard catalog
    - Create GDS user with necessary privileges in Catalog database

```
[oracle@fldcnode9]$ sqlplus / as sysdba
SQL> alter system set db_create_file_dest='/u01/app/oracle/oradata' scope=both;
SQL> alter system set open_links=16 scope=spfile;
SQL> alter system set open_links_per_instance=16 scope=spfile;
SQL> alter user gsmcatuser identified by gsmcatuser account unlock;
SQL> create user mygdsadmin identified by mygdsadmin;
SQL> grant connect, create session, gsmadmin_role to mygdsadmin;
SQL> grant inherit privileges on user SYS to GSMADMIN_INTERNAL;
```

    - Enable Trace and Restart Catalog database

```
SQL> alter system set events 'immediate trace name GWM_TRACE level 7';
SQL> alter system set event='10798 trace name context forever, level 7' s          cope=spfile;
```

    - Restart the catalog database

Oracle Sharding Deployment

- Implementation
  - Prepare Catalog database
    - Configure the Remote Scheduler

```
[oracle@fldcnode9]$ . shdcat.sh
[oracle@fldcnode9]$ sqlplus / as sysdba
SQL> @?/rdbms/admin/prvtrsch.plb
SQL> exec dbms_xdb.sethttpport(8080);
SQL> exec DBMS_SCHEDULER.SET_AGENT_REGISTRATION_PASS(‘oracleagent');
SQL> alter system register;
```

Oracle Sharding Deployment

- Implementation
  - Prepare Catalog database
    - Start scheduler and register database on SDB and all Shard Servers

```
[oracle@fldcnode10]$ . shd1.sh
[oracle@fldcnode10]$ schagent -start
Scheduler agent started using port 26871
[oracle@fldcnode10$ schagent -status
Agent running with PID 814

Agent_version:12.2.0.1.0
Running_time:00:00:13
Total_jobs_run:0
Running_jobs:0
Platform:Linux
ORACLE_HOME:/u01/app/oracle/product/12.2.0/db_1
ORACLE_BASE:/u01/app/oracle
Port:26871
Host:fldcnode10

[oracle@fldcnode10]$ echo oracleagent | schagent -registerdatabase fldcnode10
Agent Registration Password ?
Oracle Scheduler Agent Registration for 12.2.0.1.0 Agent
Agent Registration Successful!
```

Oracle Sharding Deployment

- Implementation
  - Add GSM and Start GSM

```
[oracle@fldcnode9]$ . shd_dir1.sh
[oracle@fldcnode9]$ echo $ORACLE_HOME
/u01/app/oracle/product/12.2.0/gsm_1
[oracle@fldcnode9]$ gdsctl
GDSCTL: Version 12.2.0.1.0 - Production on Wed Sep 26 18:10:46 CDT 2018
Copyright (c) 2011, 2016, Oracle.  All rights reserved.
Welcome to GDSCTL, type "help" for information
GDSCTL> connect mygdsadmin/mygdsadmin
Catalog connection is established
GDSCTL> add gsm -gsm sharddirector1 -listener 1571 -pwd gsmcatuser -catalog
fldcnode9:1521:shdcat -region region1 -trace_level 16
GSM successfully added
GDSCTL> start gsm -gsm sharddirector1
GSM is started successfully
GDSCTL> set _event 17 -config_only
Event 17 is set
GDSCTL> set gsm -gsm sharddirector1
```

Oracle Sharding Deployment

- Implementation
  - Check GSM status

```
GDSCTL> status gsm
Alias                    SHARDDIRECTOR1
Version                    12.2.0.1.0
Start Date                  26-SEP-2018 21:12:32
Trace Level                 support
Listener Log File         /u01/app/oracle/diag/gsm/fldcnode9/sharddirector1/alert/log.xml
Listener Trace File       /u01/app/oracle/diag/gsm/fldcnode9/sharddirector1/trace/ora_33205_139712583494496.trc
Endpoint summary          (ADDRESS=(HOST=fldcnode9)(PORT=1571)(PROTOCOL=tcp))
GSMOCI Version            2.2.1
Mastership                 Y
Connected to GDS catalog  Y
Process Id                33288
Number of reconnections   0
Pending tasks.     Total  0
Tasks in  process. Total  0
Regional Mastership       TRUE
Total messages published  0
Time Zone                  -05:00
Orphaned Buddy Regions: None
GDS region                region1
```

Oracle Sharding Deployment

- Implementation
  - Create Shard Catalog

```
[oracle@fldcnode9]$ . shd_dir1.sh
[oracle@fldcnode9]$ gdsctl
GDSCTL> create shardcatalog -database fldcnode9:1521:shdcat -chunks 12 -
user mygdsadmin/mygdsadmin -sdb shdcat -region region1,region2 -
agent_port 8080


Catalog is created
```

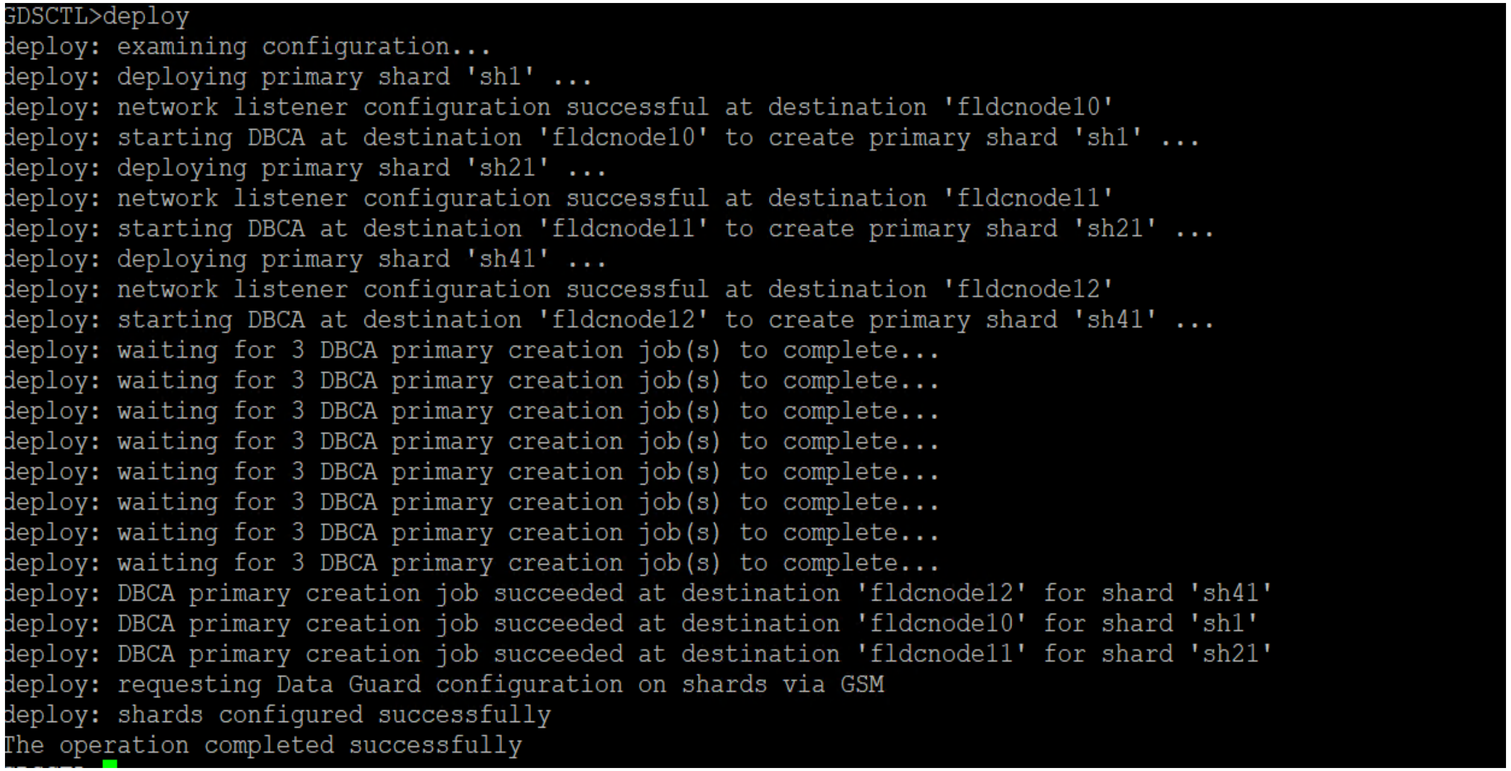
Oracle Sharding Deployment

- Implementation
  - Create Shard Groups and Shards

```
GDSCTL> add shardgroup -shardgroup primary_shardgroup -deploy_as primary -region region1
The operation completed successfully
GDSCTL> add invitednode fldcnode10
GDSCTL> create shard -shardgroup shgrp1 -destination fldcnode10 -osaccount oracle -ospassword oracle
The operation completed successfully
DB Unique Name: sh1
GDSCTL> add invitednode fldcnode11
GDSCTL> create shard -shardgroup shgrp1 -destination fldcnode11 -osaccount oracle -ospassword oracle
GDSCTL> add invitednode fldcnode12
GDSCTL> create shard -shardgroup shgrp1 -destination fldcnode12 -osaccount oracle -ospassword oracle
GDSCTL> config shard
Catalog connection is established
Name                Shard Group          Status    State       Region
Availability
sh1                 shgrp1               U         none        region1    -
sh21                shgrp1               U         none        region1    -
sh41                shgrp1               U         none        region1    -
```

Oracle Sharding Deployment

- Implementation
  - Deploy
    - Deploy invokes netca and dbca to configure listener and create database (shards) on each servers added to the SDB
    - GDSCTL> deploy



```
GDSCTL>deploy
deploy: examining configuration...
deploy: deploying primary shard 'sh1' ...
deploy: network listener configuration successful at destination 'fldcnode10'
deploy: starting DBCA at destination 'fldcnode10' to create primary shard 'sh1' ...
deploy: deploying primary shard 'sh21' ...
deploy: network listener configuration successful at destination 'fldcnode11'
deploy: starting DBCA at destination 'fldcnode11' to create primary shard 'sh21' ...
deploy: deploying primary shard 'sh41' ...
deploy: network listener configuration successful at destination 'fldcnode12'
deploy: starting DBCA at destination 'fldcnode12' to create primary shard 'sh41' ...
deploy: waiting for 3 DBCA primary creation job(s) to complete...
deploy: waiting for 3 DBCA primary creation job(s) to complete...
deploy: waiting for 3 DBCA primary creation job(s) to complete...
deploy: waiting for 3 DBCA primary creation job(s) to complete...
deploy: waiting for 3 DBCA primary creation job(s) to complete...
deploy: waiting for 3 DBCA primary creation job(s) to complete...
deploy: waiting for 3 DBCA primary creation job(s) to complete...
deploy: waiting for 3 DBCA primary creation job(s) to complete...
deploy: DBCA primary creation job succeeded at destination 'fldcnode12' for shard 'sh41'
deploy: DBCA primary creation job succeeded at destination 'fldcnode10' for shard 'sh1'
deploy: DBCA primary creation job succeeded at destination 'fldcnode11' for shard 'sh21'
deploy: requesting Data Guard configuration on shards via GSM
deploy: shards configured successfully
The operation completed successfully
```

Oracle Sharding Deployment

- Implementation
  - Deploy

GDSCTL> config shard

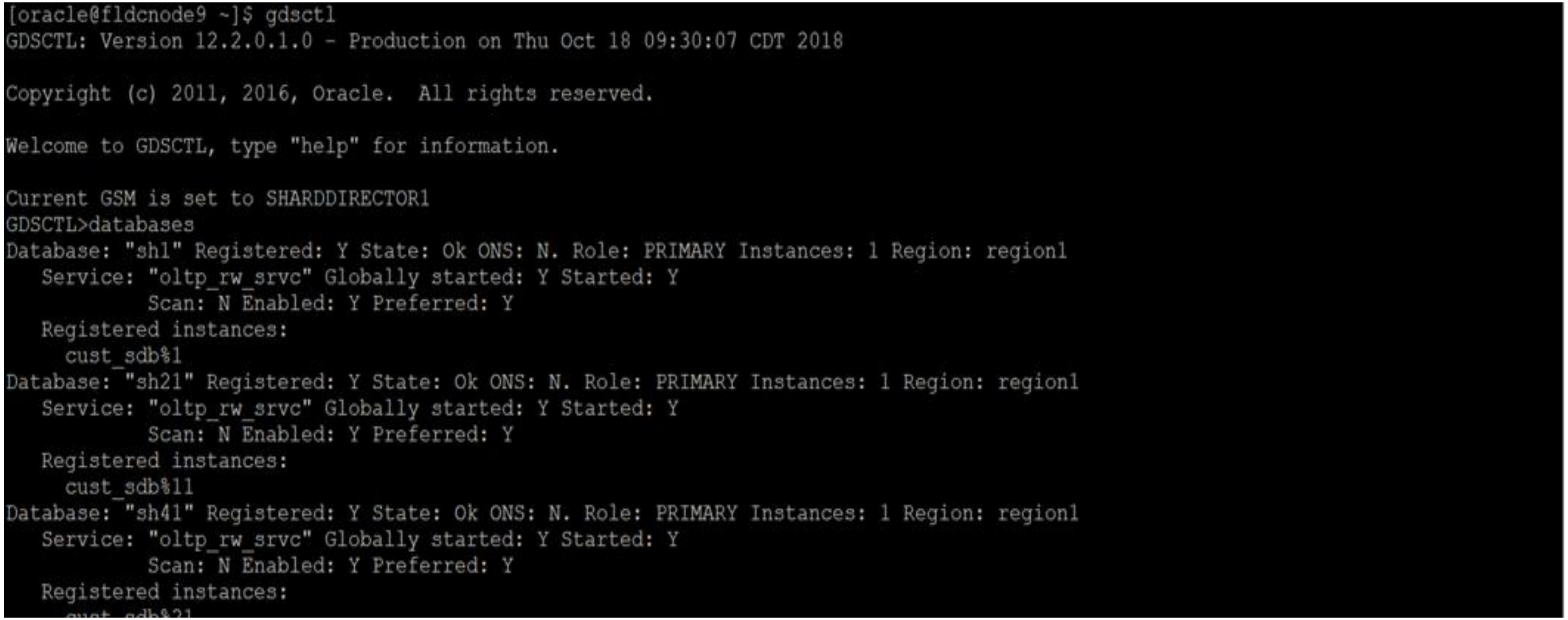
Catalog connection is established

```
Name                Shard Group         Status    State       Region
Availability
----                -----------         ------    -----       ------    --
----------
sh1                 shgrp1              Ok      Deployed    region1
Online
sh21                shgrp1              Ok      Deployed    region1
Online
sh41                shgrp1              Ok      Deployed    region1
Online
```

Oracle Sharding Deployment

- Implementation
  - Verify all Shards are registered
  - $gdsctl

  GDSctl>databases



```
[oracle@fldcnode9 ~]$ gdsctl
GDSCTL: Version 12.2.0.1.0 - Production on Thu Oct 18 09:30:07 CDT 2018

Copyright (c) 2011, 2016, Oracle.  All rights reserved.

Welcome to GDSCTL, type "help" for information.

Current GSM is set to SHARDDIRECTOR1
GDSCTL>databases
Database: "sh1" Registered: Y State: Ok ONS: N. Role: PRIMARY Instances: 1 Region: region1
   Service: "oltp_rw_srvc" Globally started: Y Started: Y
            Scan: N Enabled: Y Preferred: Y
   Registered instances:
     cust_sdb%1
Database: "sh21" Registered: Y State: Ok ONS: N. Role: PRIMARY Instances: 1 Region: region1
   Service: "oltp_rw_srvc" Globally started: Y Started: Y
            Scan: N Enabled: Y Preferred: Y
   Registered instances:
     cust_sdb%11
Database: "sh41" Registered: Y State: Ok ONS: N. Role: PRIMARY Instances: 1 Region: region1
   Service: "oltp_rw_srvc" Globally started: Y Started: Y
            Scan: N Enabled: Y Preferred: Y
   Registered instances:
     cust_sdb%21
```

Oracle Sharding Deployment

- Implementation
  - Create and Start Global Services

```
GDSCTL>add service -service prim_srv -role primary
GDSCTL>add service -service oltp_rw_srvc -role primary
The operation completed successfully
GDSCTL>config service
Name           Network name                      Pool           Started Preferred all
----           ------------                      ----           ------- -------------
oltp_rw_srvc   oltp_rw_srvc.cust_sdb.oradbcloud  cust_sdb       No      Yes

GDSCTL>start service -service oltp_rw_srvc
The operation completed successfully
GDSCTL>config service
Name           Network name                      Pool           Started Preferred all
----           ------------                      ----           ------- -------------
oltp_rw_srvc   oltp_rw_srvc.cust_sdb.oradbcloud  cust_sdb           Yes      Yes

GDSCTL>status service
Service "oltp_rw_srvc.cust_sdb.oradbcloud" has 3 instance(s). Affinity: ANYWHERE
  Instance "cust_sdb%1", name: "sh1", db: "sh1", region: "region1", status: ready.
  Instance "cust_sdb%11", name: "sh21", db: "sh21", region: "region1", status: ready.
  Instance "cust_sdb%21", name: "sh41", db: "sh41", region: "region1", status: ready.
```

Oracle Sharding Deployment

- Implementation
  - Create Application User/Schema with Necessary Privileges

```
SQL> alter session enable shard ddl;

SQL> create user apps identified by apps;

SQL> grant connect, resource, alter session to apps;

SQL> grant execute on dbms crypto to apps;

SQL> grant create table, create procedure, create tablespace, create materialized view to app schema;

SQL> grant unlimited tablespace to apps;

SQL> grant select catalog role to apps;

SQL> grant all privileges to apps;

SQL> grant gsmadmin role to apps;

SQL> grant dba to apps;
```

Oracle Sharding Deployment

- Implementation
  - Create TABLESPACE SET for SHARDED TABLES

```
SQL> CREATE TABLESPACE SET TBS_SET_1 USING TEMPLATE
(DATAFILE SIZE 100M AUTOEXTEND ON
NEXT IOM MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL SEGMENT
 SPACE MANAGEMENT AUTO) ;

SQL> CREATE TABLESPACE SET TBS_SET_2
DATAFILE SIZE 100M AUTOEXTEND ON
NEXT IOM MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL
UNIFORM SIZE 1M;
```

Oracle Sharding Deployment

- Implementation
  - Create Sharded Tables and Duplicated Tables

```
SQL> alter session enable shard ddl;
SQL> connect apps/ apps
SQL> CREATE SHARDED TABLE Customers
(custId VARCHAR2 (60) NOT NULL, First-Name VARCHAR2 (60), Last-Name VARCHAR2 (60), Class
VARCHAR2 (10), Geo VARCHAR2 (8), CustProfi1e VARCHAR2 (4000), Passwd RAW (60),
CONSTRAINT pk customers PRIMARY KEY (CustId),
CONSTRAINT json customers CHECK (Custprofile IS JSON) ) TABLESPACE SET TSP_SET_1
PARTITION BY CONSISTENT HASH (CustId) PARTITIONS AUTO;


CREATE SHARDED TABLE LineItems
(OrderId INTEGER NOT NULL, CustId VARCHAR2(60) NOT NULL,
ProductId INTEGER NOT NULL SYSTEM MANAGED SDB WITH ACTIVE DATA GUARD
Price NUMBER(19,4), Qty NUMBER,
constraint pk_items primary key (CustId, OrderId, ProductId),
constraint fk_items_parent foreign key (CustId, OrderId)
references Orders on delete cascade
```

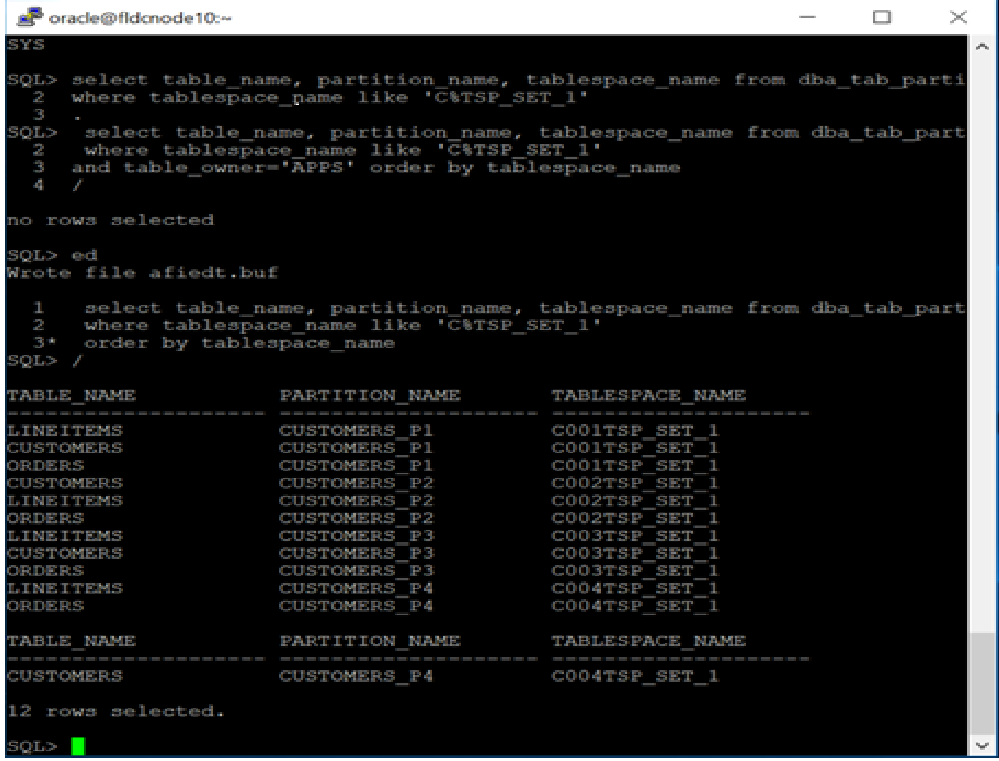
Oracle Sharding Deployment

- Implementation
  - Create Sharded Tables and Duplicated Tables

```
SQL> CREATE DUPLICATED TABLE Products
(ProductId INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
Name VARCHAR2(128),
DescrUri VARCHAR2(128),
LastPrice NUMBER(19,4)
) TABLESPACE products_tsp
```

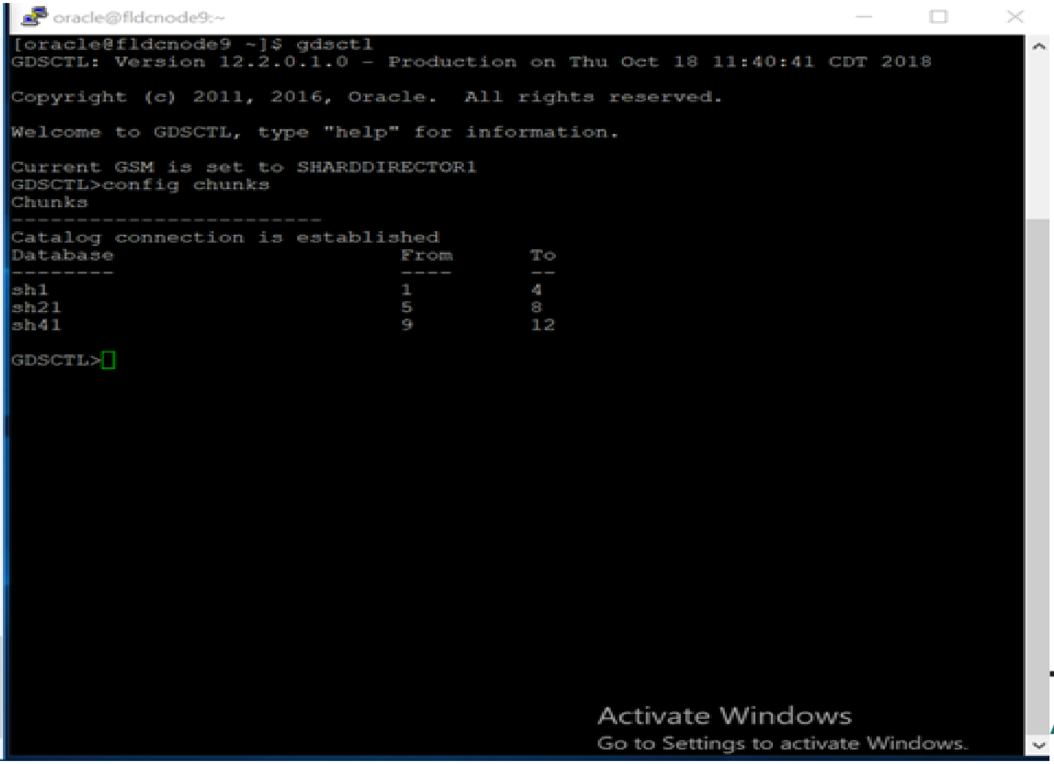
Oracle Sharding Deployment

- Implementation
  - Distribution of tables across Shards
  - The left screen shot shows the table partition(chunk) in one shard:
  - 4 Tablespaces: C001TSP_SET_1 to C004-TSPSET_1
  - 4 Chunks: CUTOMMERS_F1 to CUTOMMERS_F4
  - 4 tables in the shard family



oracle@fldcnode10:~

```
SYS

SQL> select table_name, partition_name, tablespace_name from dba_tab_parti
  2  where tablespace_name like 'C%TSP_SET_1'
  3  .
SQL>  select table_name, partition_name, tablespace_name from dba_tab_part
  2   where tablespace_name like 'C%TSP_SET_1'
  3  and table_owner='APPS' order by tablespace_name
  4  /

no rows selected

SQL> ed
Wrote file afiedt.buf

  1   select table_name, partition_name, tablespace_name from dba_tab_part
  2   where tablespace_name like 'C%TSP_SET_1'
  3*  order by tablespace_name
SQL> /

TABLE_NAME                PARTITION_NAME            TABLESPACE_NAME
------------------------- ------------------------- -------------------------
LINEITEMS                 CUSTOMERS_P1              C001TSP_SET_1
CUSTOMERS                 CUSTOMERS_P1              C001TSP_SET_1
ORDERS                    CUSTOMERS_P1              C001TSP_SET_1
CUSTOMERS                 CUSTOMERS_P2              C002TSP_SET_1
LINEITEMS                 CUSTOMERS_P2              C002TSP_SET_1
ORDERS                    CUSTOMERS_P2              C002TSP_SET_1
LINEITEMS                 CUSTOMERS_P3              C003TSP_SET_1
CUSTOMERS                 CUSTOMERS_P3              C003TSP_SET_1
ORDERS                    CUSTOMERS_P3              C003TSP_SET_1
LINEITEMS                 CUSTOMERS_P4              C004TSP_SET_1
ORDERS                    CUSTOMERS_P4              C004TSP_SET_1

TABLE_NAME                PARTITION_NAME            TABLESPACE_NAME
------------------------- ------------------------- -------------------------
CUSTOMERS                 CUSTOMERS_P4              C004TSP_SET_1

12 rows selected.

SQL> 
```



oracle@fldcnode9:~

```
[oracle@fldcnode9 ~]$ gdsctl
GDSCTL: Version 12.2.0.1.0 - Production on Thu Oct 18 11:40:41 CDT 2018

Copyright (c) 2011, 2016, Oracle.  All rights reserved.

Welcome to GDSCTL, type "help" for information.

Current GSM is set to SHARDDIRECTOR1
GDSCTL>config chunks
Chunks
------------------------
Catalog connection is established
Database                          From      To
--------                          ----      --
sh1                               1         4
sh21                              5         8
sh41                              9         12

GDSCTL>
```

Introduction to a sharding project

- Global customer order applications.
    - Database : global customer information, order information, etc.
    - Consolidate databases of business units: current layout:
        - 3 separate databases for regions: Americas, EMEA, APJ
        - One schema per country in each database
        - Same tables structure among schemas.
        - Applications for different countries to connect to difference schema.
- Candidate for sharding
    - Current architecture has limitation on scalability and availability
    - Potential candidate for using Oracle sharding to improve scalability and fault isolation
- Sharding architecture design
    - Add region column to some of sharded tables:

      shard tables: customer table and its associated tables

      Region code: List values: (‘americas’, ‘emea’, ‘apj’)

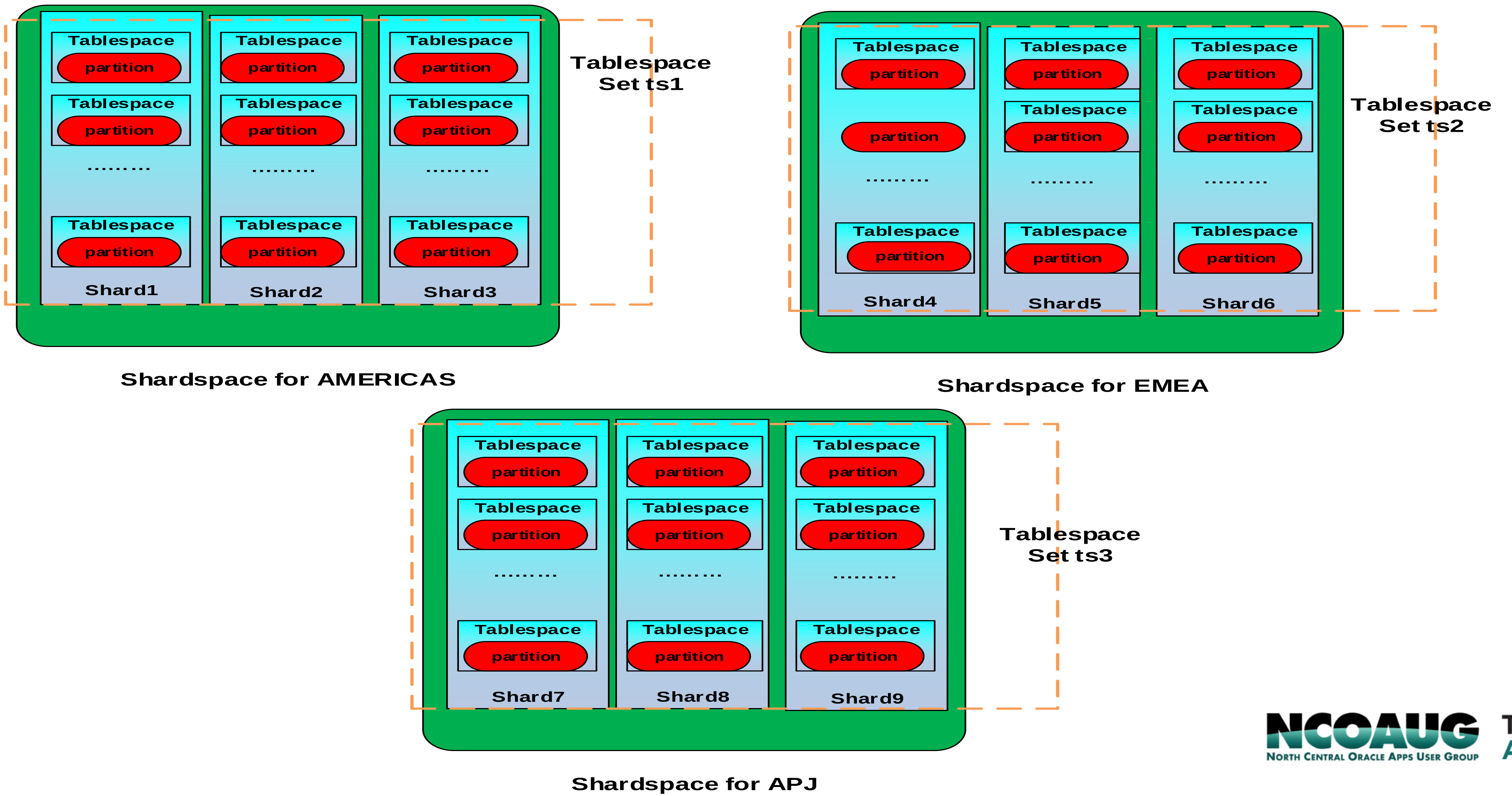
Introduction to a Sharding Project

- Sharding architecture design
  - Sharding method: composite sharding, two levels of sharding:
    - User-defined sharding by region code as sharding key
    - System-managed shard uses consistency hash on custid within the region.
  - Customer table:

```
 CREARE SHARD TABLE customers
(custid   NUMBER  NOT NULL,    custname           VARCHAR(50),
 currencycode    VARCHAR(3),     address            VARCHAR(100),
 countrycode      VARCHAR(10),   regioncode         VARCHAR(10),
 CONSTRAINT cust_pk PRIMARY KEY(custid, regioncode))
 PARTITION BY LIST (regioncode)
 PARTITION AUTO
 (PARTITIONSET AMERICAS VALUES ('americas') TABLESPACE SET tbs1,
  PARTITIONSET EMEA VALUES('emea') TABLESPACE SET tbs2,
   PARTITIONSET APJ VALUES('apj') TABLESPACE SET tbs3,
```

  - Nine shards: shard1-3 for AMERICAS Region, shard 4-6 for EMEA Region.
    shard 7-9 for APJ Region.

Shard Databases Architecture Design



References:

1. System Managed Sharding with Active Data Guard Using create shard method, cookbook, Sept 2017

2. Oracle Database using Oracle sharding, 19c, E87088-04 February 2019

3. Oracle Sharding 18c New Features, **Oracle whitepaper| July 2018**

Thank You and QA

Contact me at kai_yu@dell.com or visit my Oracle Blog at http://kyuoracleblog.wordpress.com/