



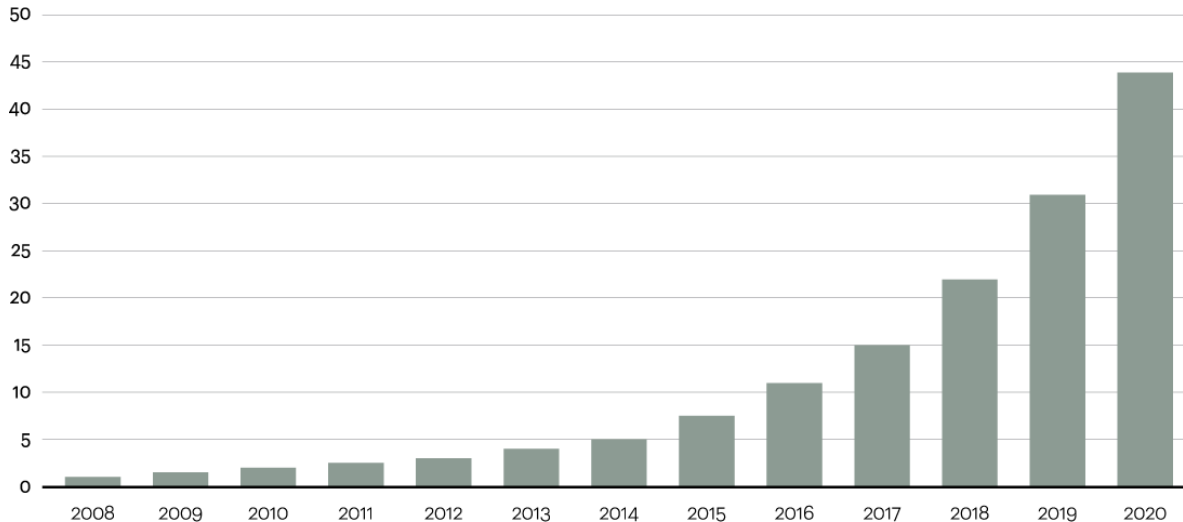
Welcome rebel leaders

Data's massive growth in last 20 years

Figure 1

Data is growing at a 40 percent compound annual rate, reaching nearly 45 ZB by 2020

Data in zettabytes (ZB)



Source: Oracle, 2012



The old republic

74



65



53



Just keep building it bigger



Eventually it was just too much



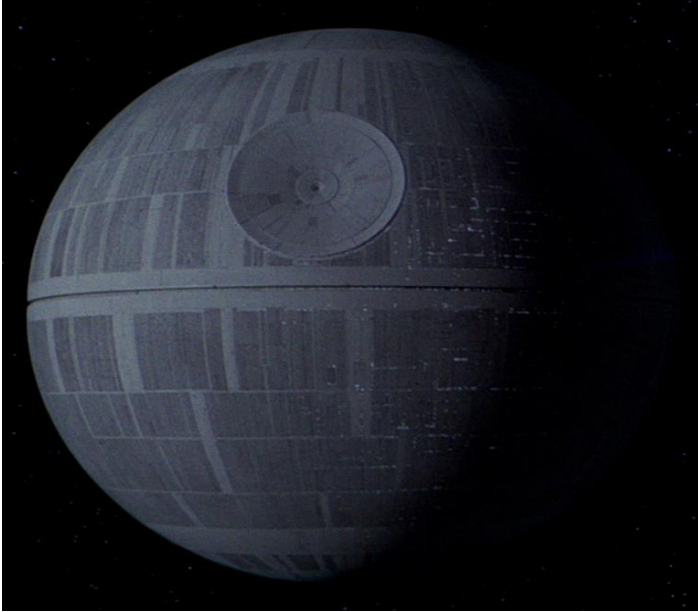


May the Shard be with you

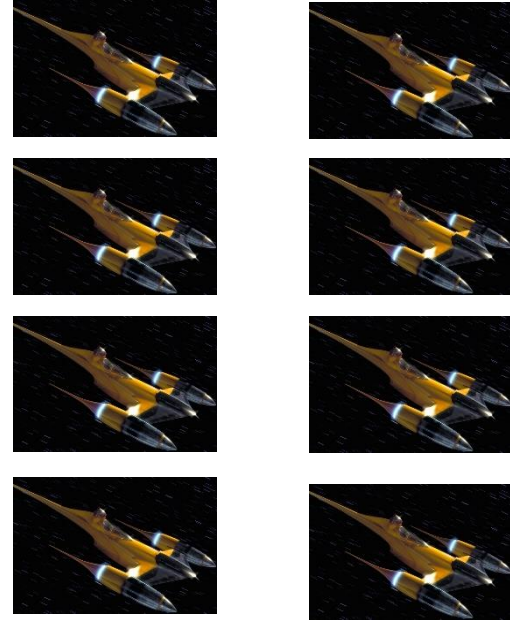
Sharding

- Sharding?
 - Scale out not up
 - Dividing large tables in smaller more manageable tables.
 - Horizontal partitioning by row on a larger scale.
 - Each database shard can be on its own server to spread the load
 - Lets you run many less expensive commodity servers rather than one expensive server
 - By splitting tables into shards the processing workload gets spread around each shard, rather than overwhelming a single server.
 - Design of the database schemas and applications use is critical to getting the benefits of sharding

Sharding. Save \$\$\$



Instead of one huge resource
consuming server



Many smaller commodity
servers

May the Shard Key be with you

- Shard Key
 - The shard key determines where your data will go AND will affect the over all performance of your shards and shard cluster
 - Take time to look and understand your data and how it will be used before you decide on a key
 - The optimal key will allow your data to be spread evenly amongst all your nodes
 - High cardinality – allows for good sharding but does not guarantee equal data distribution
 - Frequency of data – low frequency is better but again does not guarantee equal data distribution
 - Rate of change – do not use a key that increases monotonically.
- The perfect key
 - All update, inserts, deletes will be evenly distributed
 - All operations would only go to the shard where the data resides
 - A query would never ask for data from a shard that doesn't have the data needed
- The perfect shard key is like the holy grail...it doesn't exist. There is always some compromise. Again KNOW how your data is used and design the best key for your needs

The dark side ...

- Hotspots

- uneven distribution of data. One shard stays busier than all the other shards. A hotspot can completely destroy the benefits of sharding



A bad shard key will...

- Cross Partition operations

- Cause operations to search through some or even all shards to find the requested data.
- Again this can defeat the purpose of sharding



Range Sharding

- Maps the shard key value directly to a shard/server.
- Easy to setup and conceptually simple
- Should not use a monotonically increasing number (ex. timestamp) as a shard key
 - Your inserts will always happen on the shard that holds the max value.
- Can reduce scalability and performance.
 - If you add or remove shards, the amount of data to be relocated may affect performance or even require taking the system offline while data is being reconfigured.

Range Partition example

CUST_ID	LNAME	FNAME	ADDRESS	STATE	ZIPCODE	PHONE	EMAIL
1	Smith	Jane	123 her street	CA	91327	352-761-0962	jsmith@email.com
2	Doe	John	123 his street	MA	02112	756-765-1234	jdoe@email.com
3	Abbey	Ralph	333 his street	WA	98105	345-231-2345	rabbey@email.com
4	Rogers	Tom	645 his street	MS	38601	837-352-0097	trogers@email.com
5	Marks	Mary	451 her street	UT	84101	726-444-1928	mmarks@email.com
6	Perkins	Grace	998 her street	FL	32830	234-927-1349	gperkins@email.com

Shard 1

1	Smith	Jane	123 her street	CA	91327	352-761-0962	jsmith@email.com
2	Doe	John	123 his street	MA	02112	756-765-1234	jdoe@email.com

Shard 2

3	Abbey	Ralph	333 his street	WA	98105	345-231-2345	rabbey@email.com
4	Rogers	Tom	645 his street	MS	38601	837-352-0097	trogers@email.com

Shard 3

5	Marks	Mary	451 her street	UT	84101	726-444-1928	mmarks@email.com
6	Perkins	Grace	998 her street	FL	32830	234-927-1349	gperkins@email.com

Shard max

n+1							

List Partitioning

- Maps the shard key values to a list of discreet values.
- Easy to setup
- Allows you to group and organize your data in a meaningful way
- Can give improved performance since you may only need to query one shard for your data

List partitioning example

CUST_ID	LNAME	FNAME	ADDRESS	STATE	ZIPCODE	PHONE	EMAIL
1	Smith	Jane	123 her street	CA	91327	352-761-0962	jsmith@email.com
2	Doe	John	123 his street	MA	02112	756-765-1234	jdoe@email.com
3	Abbey	Ralph	333 his street	WA	98105	345-231-2345	rabbey@email.com
4	Rogers	Tom	645 his street	MS	38601	837-352-0097	trogers@email.com
5	Marks	Mary	451 her street	UT	84101	726-444-1928	mmarks@email.com
6	Perkins	Grace	998 her street	FL	32830	234-927-1349	gperkins@email.com

Shard 1 NW

3	Abbey	Ralph	333 his street	WA	98105	345-231-2345	rabbey@email.com
---	-------	-------	----------------	----	-------	--------------	--

Shard 2 SW

1	Smith	Jane	123 her street	CA	91327	352-761-0962	jsmith@email.com
5	Marks	Mary	451 her street	UT	84101	726-444-1928	mmarks@email.com

Shard 3 NE

2	Doe	John	123 his street	MA	02112	756-765-1234	jdoe@email.com
---	-----	------	----------------	----	-------	--------------	--

Shard 4 SE

4	Rogers	Tom	645 his street	MS	38601	837-352-0097	trogers@email.com
6	Perkins	Grace	998 her street	FL	32830	234-927-1349	gperkins@email.com

Hashed Sharding

- Provides a more even distribution of data across the shards.
- Uses a hash function on the key to decide what shard data will go into
 - the bucket number is calculated as $\text{HF}(\text{key}) \% N$ where HF is a hash function and N is the number of buckets. This approach works fine if N is constant, but requires reshuffling of all data when N changes
- Choose a key that has high cardinality or large number of different values
- Easier to avoid hotspots
- Allows easier scaling and availability for writes, BUT, query performance can be affected because there is no query isolation. You may need to query all nodes to get your data back

Hash Sharding examples

CUST_ID	LNAME	FNAME	ADDRESS	STATE	ZIPCODE	PHONE	EMAIL
1	Smith	Jane	123 her street	CA	91327	352-761-0962	jsmith@email.com
2	Doe	John	123 his street	MA	02112	756-765-1234	jdoe@email.com
3	Abbey	Ralph	333 his street	WA	98105	345-231-2345	rabbey@email.com
4	Rogers	Tom	645 his street	MS	38601	837-352-0097	trogers@email.com
5	Marks	Mary	451 her street	UT	84101	726-444-1928	mmarks@email.com
6	Perkins	Grace	998 her street	FL	32830	234-927-1349	gperkins@email.com

Shard 1

1	Smith	Jane	123 her street	CA	91327	352-761-0962	jsmith@email.com
4	Rogers	Tom	645 his street	MS	38601	837-352-0097	trogers@email.com

Shard 2

3	Abbey	Ralph	333 his street	WA	98105	345-231-2345	rabbey@email.com
6	Perkins	Grace	998 her street	FL	32830	234-927-1349	gperkins@email.com

Shard 3

2	Doe	John	123 his street	MA	02112	756-765-1234	jdoe@email.com
5	Marks	Mary	451 her street	UT	84101	726-444-1928	mmarks@email.com

Round Robin Sharding

- Data is spread around the shards in a round robin method.
- You get equal data distribution among all the shards
- Adding a shard can be problematic since new shards will not have the same amount of data unless you rebalance everything.
- Query performance will be affected just like with hash sharding

Shard 1

1Smith	Jane	123 her street	CA	91327	352-761-0962	jsmith@email.com
4Rogers	Tom	645 his street	MS	38601	837-352-0097	trogers@email.com

Shard 2

2Doe	John	123 his street	MA	02112	756-765-1234	jdoe@email.com
5Marks	Mary	451 her street	UT	84101	726-444-1928	mmarks@email.com

Shard 3

3Abbey	Ralph	333 his street	WA	02108	345-231-2345	rabbey@email.com
6Perkins	Grace	998 her street	FL	32830	234-927-1349	gperkins@email.com

Sharding Cons: Beware the Dark Side

- Joins and Denormalization
 - Often not feasible since joins would require accessing data across different shards
- Referential integrity
 - Foreign keys are very hard to manage in a shared environment. Usually have to have the application enforce them.
- Once a shard key is implemented, it is immutable.
 - Make sure the shard key you create is the one you need
 - If you ever have to change the key, you would have to do a rebalance which usually requires all the data to be moved and that means downtime.
- Not a good use for small tables
- Share nothing technology
 - This is not bad, BUT you have to make sure you have redundancy of the nodes in case one goes down. Can also use the node replicas to spread the load even more

Oracle sharding



- New in 12.2
- Data is horizontally partitioned across independent databases collectively called a SDB
- Based on the shared nothing architecture and eliminates a single point of failure
- Data Guard can be used for DR and HA for each node
- Sharding methods
 - System Managed Sharding
 - Composite Sharding
 - Subpartitions with Sharding

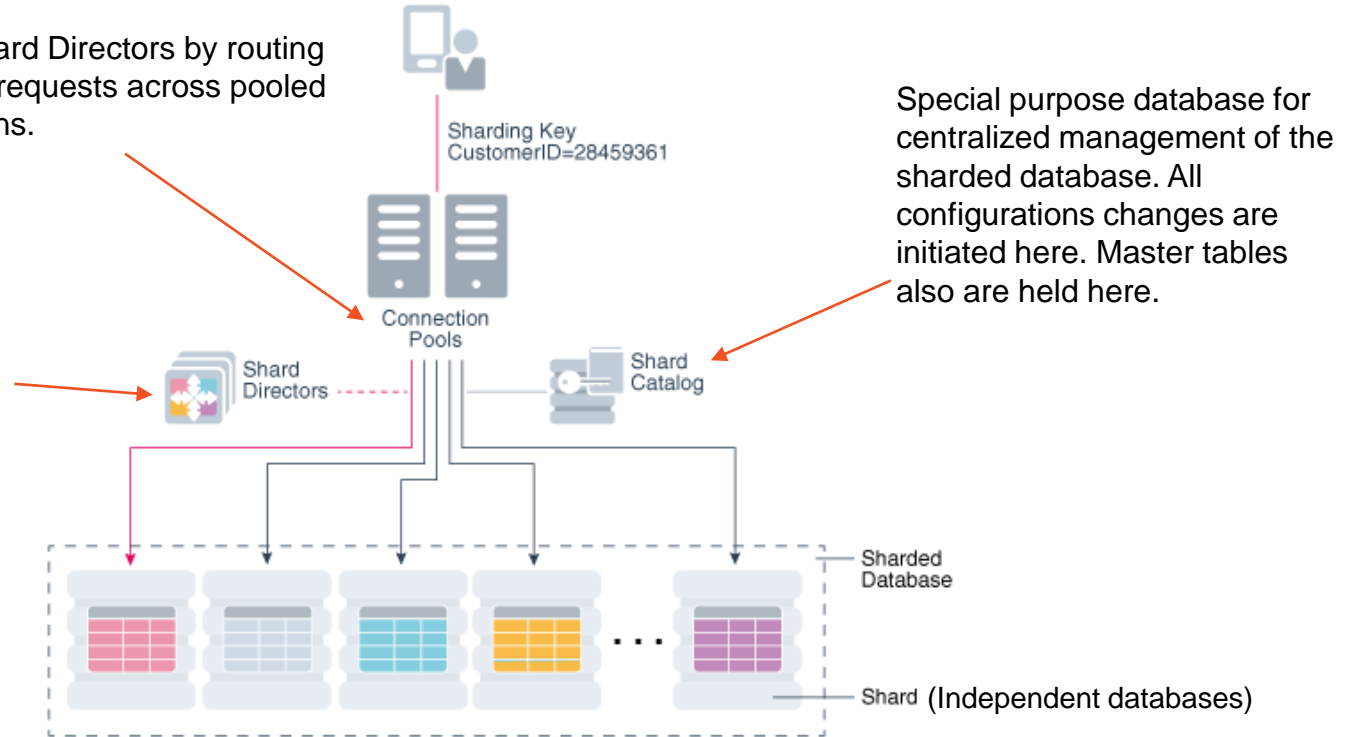
Provides the benefits of sharding without sacrificing

- Relational schemas
- Database partitioning
- ACID properties and read consistency
- SQL and other programmatic interfaces
- Complex data types
- Online schema changes
- Multi-core scalability
- Advanced security
- Compression
- High Availability features
- Enterprise-scale backup and recovery

Architecture

Act as Shard Directors by routing database requests across pooled connections.

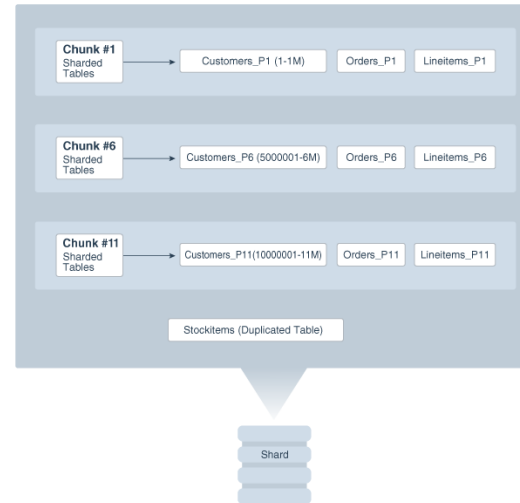
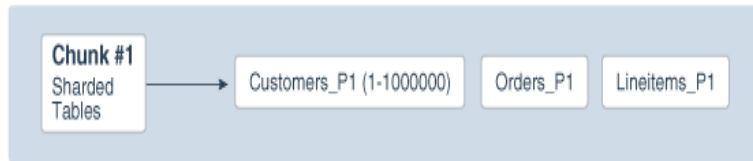
Specific implementation of a global service manager
The director maintains a current topology map of the SDB.
Based on the sharding key passed during a connection request, the director routes the connections to the appropriate shard.
For a typical SDB, a set of shard directors are installed on dedicated low-end commodity servers in each region.
To achieve high availability, deploy multiple shard directors.



<https://docs.oracle.com/en/database/oracle/oracle-database/12.2/admin/sharding-overview.html#GUID-3D41F762-BE04-486D-8018-C7A210D809F9>

Chunks and Shards

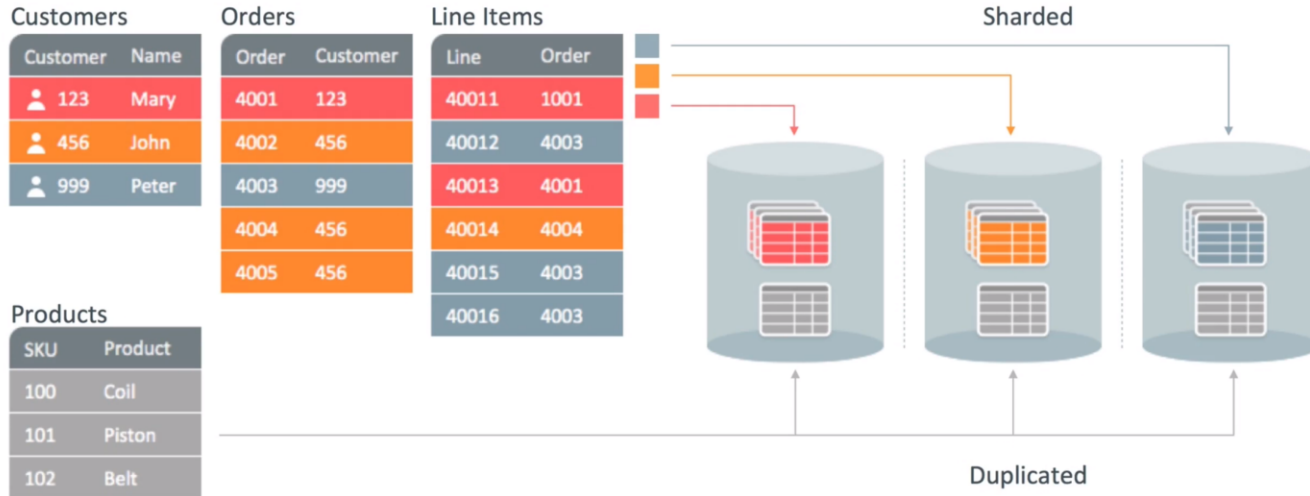
- Each partition of a sharded table is stored in a separate tablespace.
- Its best to group similar tablespaces that have similar partitions for tables together to a chunk
- A chunk is a group of partitions from different tables that correspond to each other
- A shard can contain more than one chunk
- Duplicate tables(mat views) also are in shards



<https://docs.oracle.com/en/database/oracle/oracle-database/12.2/admin/sharding-physical-organization.html#GUID-3D0475BE-3F6F-4D2E-8B2E-B1161DBB1A78>

Chunks and Shards

Sharded Schema



<https://www.youtube.com/watch?v=ACXeZCOo4uU>

Oracle sharding methods

- System-managed Sharding
 - User does not need to specify any mapping
 - Uses consistent hash to randomly and evenly spread data across nodes
 - Avoids hot spots
 - Oracle will automatically maintain a balance of data as nodes are added or removed from a SDB
- Composite Sharding
 - Provides 2 levels of sharding
 - High level by list or range using a shard key based on the list or range
 - Then partitioned again by consistent hash using a normal shard key
 - SDB is divided into multiple sets of shards called shardspaces
 - Good for geographic distribution scenarios
- Subpartitions and sharding
 - Since Oracle sharding uses its partitioning ability, you can use all the sub partitioning methods provided by Oracle
- User defined
 - Lets you explicitly specify the mapping of your data. This could be for performance or regulatory reasons



SharePlex- Horizontal Partitioning

Your solution to control the Datas.

SharePlex Horizontal Partitioning



Source Database
Customer table

Last name A-I



Shard Blue

Last name J-S



Shard Gold

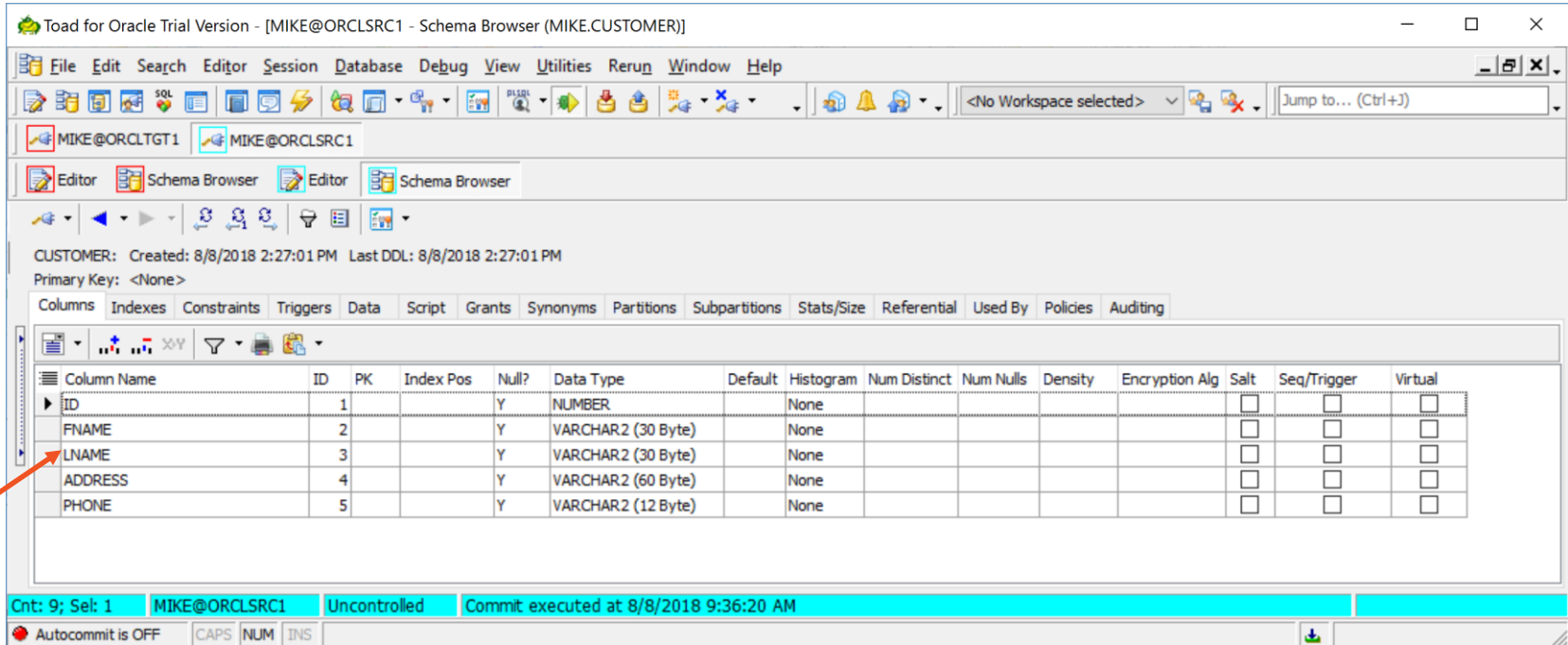
Last name T-Z



Shard Red

Setting it up

Source Table MIKE.CUSTOMER



Toad for Oracle Trial Version - [MIKE@ORCLSRC1 - Schema Browser (MIKE.CUSTOMER)]

File Edit Search Editor Session Database Debug View Utilities Rerun Window Help

MIKE@ORCLTGT1 MIKE@ORCLSRC1

Editor Schema Browser Editor Schema Browser

CUSTOMER: Created: 8/8/2018 2:27:01 PM Last DDL: 8/8/2018 2:27:01 PM
Primary Key: <None>

Columns Indexes Constraints Triggers Data Script Grants Synonyms Partitions Subpartitions Stats/Size Referential Used By Policies Auditing

Column Name	ID	PK	Index Pos	Null?	Data Type	Default	Histogram	Num Distinct	Num Nulls	Density	Encryption Alg	Salt	Seq/Trigger	Virtual
ID	1			Y	NUMBER		None					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FNAME	2			Y	VARCHAR2 (30 Byte)		None					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LNAME	3			Y	VARCHAR2 (30 Byte)		None					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ADDRESS	4			Y	VARCHAR2 (60 Byte)		None					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PHONE	5			Y	VARCHAR2 (12 Byte)		None					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cnt: 9; Sel: 1 MIKE@ORCLSRC1 Uncontrolled Commit executed at 8/8/2018 9:36:20 AM

Autocommit is OFF CAPS NUM INS

Setting it up cont.

Target Tables in Schemas shard1, shard2, shard3

The screenshot shows the Toad for Oracle Schema Browser interface. The left pane displays the 'SHARD1' schema with a 'Tables' list containing 'CUSTOMER'. A red arrow points from the 'CUSTOMER' table in the list to the table's details in the right pane. The right pane shows the table's metadata: 'CUSTOMER: Created: 8/8/2018 2:28:41 PM Last DDL: 8/8/2018 2:28:41 PM' and 'Primary Key: <None>'. Below this is a table of columns with the following data:

Column Name	ID	PK	Index Pos	Null?	Data Type	Default	Histogram	Num Distinct	Num
ID	1			Y	NUMBER		None		
FNAME	2			Y	VARCHAR2 (30 Byte)		None		
LNAME	3			Y	VARCHAR2 (30 Byte)		None		
ADDRESS	4			Y	VARCHAR2 (60 Byte)		None		
PHONE	5			Y	VARCHAR2 (12 Byte)		None		

The bottom status bar shows 'Cnt: 1; Sel: 1', 'MIKE@ORCLTGT1', 'Uncontrolled', and 'Autocommit is OFF'.

Setting it up cont.

Define/verify your partition schemas

add partition to schema1 set name = q1 and condition = "LNAME<'J%'" and route = 10.0.1.6:q1@o.ORCL and tablename = SHARD1.CUSTOMER

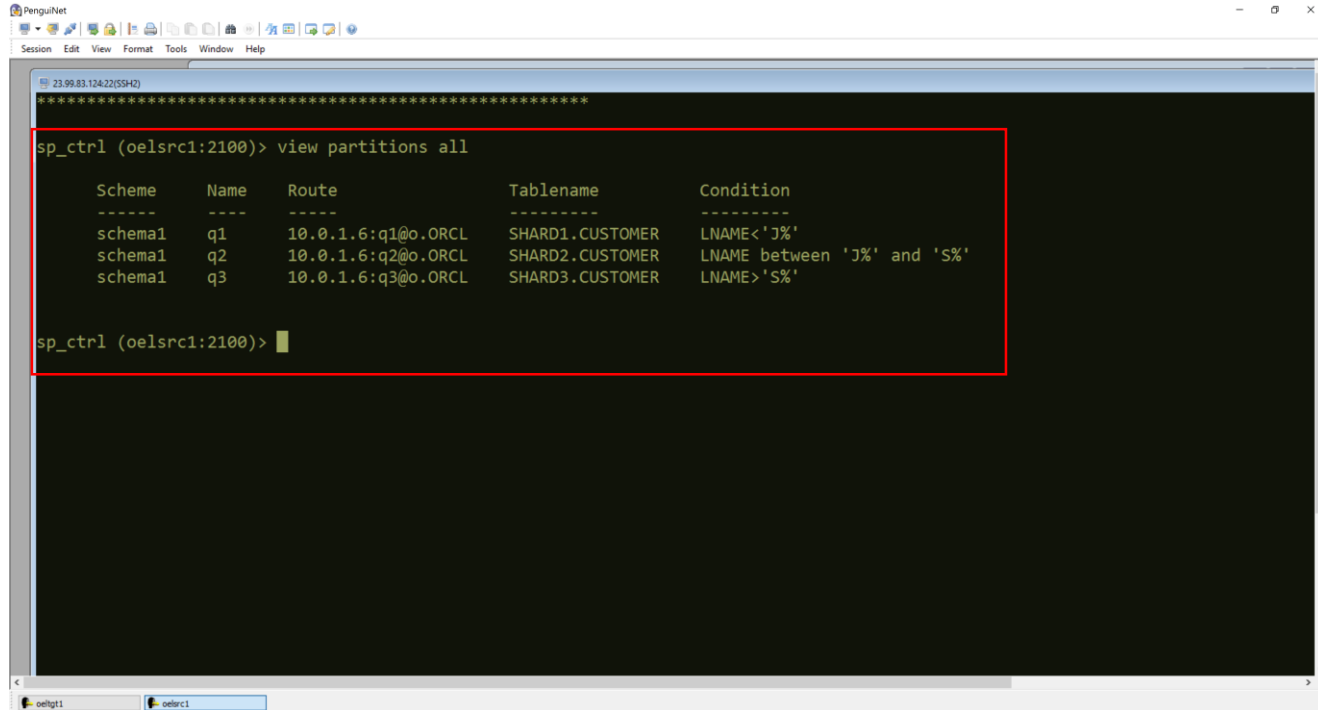
add partition to schema1 set name = q2 and condition = "LNAME between 'J%' and 'S%'" and route = 10.0.1.6:q2@o.ORCL and tablename = SHARD2.CUSTOMER

add partition to schema1 set name = q3 and condition = "LNAME>'S%'" and route = 10.0.1.6:q3@o.ORCL and tablename = SHARD3.CUSTOMER

View partitions all

Setting it up cont.

Define your partition schemas



```
PenguinNet
Session Edit View Format Tools Window Help
23.99.83.124:22(SSHD)
*****
sp_ctrl (oelsrc1:2100)> view partitions all

  Scheme      Name      Route              Tablename          Condition
  -----
  schema1     q1       10.0.1.6:q1@o.ORCL SHARD1.CUSTOMER    LNAME<'J%'
  schema1     q2       10.0.1.6:q2@o.ORCL SHARD2.CUSTOMER    LNAME between 'J%' and 'S%'
  schema1     q3       10.0.1.6:q3@o.ORCL SHARD3.CUSTOMER    LNAME>'S%'

sp_ctrl (oelsrc1:2100)> █
```

Setting it up cont.

Create your configuration file

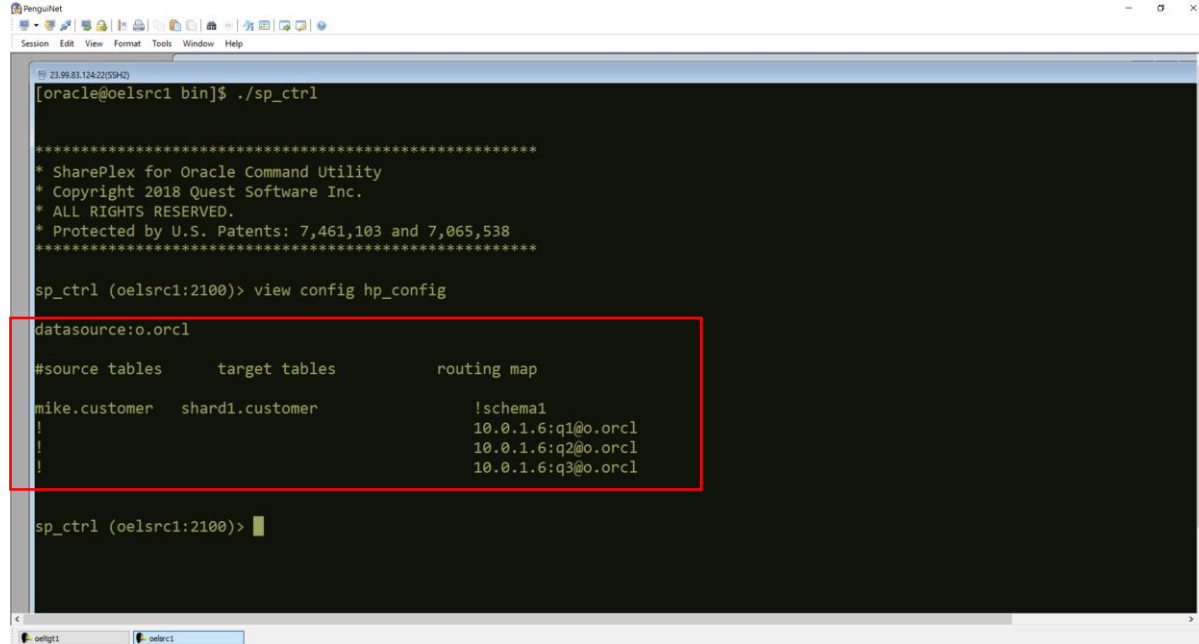
datasource:o.orcl

#source tables	target tables	routing map
----------------	---------------	-------------

mike.customer	shard1.customer	!schema1
!		10.0.1.6:q1@o.orcl
!		10.0.1.6:q2@o.orcl
!		10.0.1.6:q3@o.orcl

Setting it up cont.

Create your configuration file



```
[oracle@oelsrc1 bin]$ ./sp_ctrl

*****
* SharePlex for Oracle Command Utility
* Copyright 2018 Quest Software Inc.
* ALL RIGHTS RESERVED.
* Protected by U.S. Patents: 7,461,103 and 7,065,538
*****

sp_ctrl (oelsrc1:2100)> view config hp_config

datasource:o.orcl

#source tables      target tables      routing map
mike.customer      shard1.customer    !schema1
!                  !                  10.0.1.6:q1@o.orcl
!                  !                  10.0.1.6:q2@o.orcl
!                  !                  10.0.1.6:q3@o.orcl

sp_ctrl (oelsrc1:2100)>
```



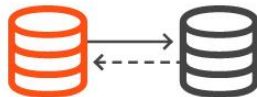
Demo

Use Cases Supported with Oracle Replication

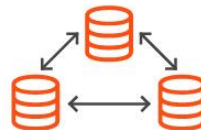
Availability



High availability/
Disaster recovery



Migrations, patches
and upgrades

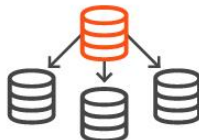


Active-Active/
Load balancing

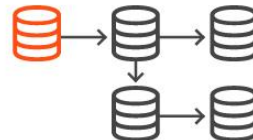
Scalability



Operational reporting/
Archiving



Data distribution/
Distributed processing



Cascading using
intermediary systems

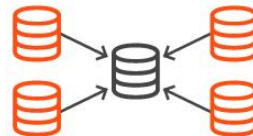
Integration



Data integration/
Data warehousing



Change history/
Metadata repository



Centralized reporting/
Consolidation

Why SharePlex – Database Replication



**Enterprise ready,
heterogeneous near
real-time database
replication**



**Comprehensive
Solution for all
replication needs**



**Unrivalled, award
winning support**

SharePlex is the best-in class enterprise solution for high availability, scalability, data integration and reporting through near real-time database replication

SharePlex is DBTA trend-setting product in data and information management and Best Streaming Data Solution, Best Data Capture Solution



Awards and Recognition

2017 Confrimit ACE Award (8x Winner)

- Winners are recognized as leaders using the “Voice of the Customer” and have established, outstanding programs to measure feedback and act upon it to increase satisfaction, improve operational efficiencies, and enhance revenue growth
- Email based survey sent after each case is closed to provide feedback on product support, support engineer and Support Portal
- 94% for overall satisfaction with support and support personnel



2017 ASP Top 10 Best Web Support Sites (4x Winner)

- Showcases excellence in online service and support
- Evaluated on overall usability, design, and navigation; Knowledge Base and search implementation; Interactive features; Community engagement; Major site development challenge



2016 KM World KM Reality Award Finalist

- Acknowledged for excellence in providing comprehensive and robust knowledge base content for our customers





SharePlex

Explore www.SharePlex.com

- Contact your Sales Rep.
- Phil.Rodas@quest.com
- [Join the SharePlex Community](#)



Quest[™]