PassQueen
a.k.a dumpsqueen.com

MySQL HeatWave Implementation Associate Rel 1

Oracle 1Z0-915-1

Version Demo

Total Demo Questions: 7

Total Premium Questions: 76

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, MySQL Database Service	29
Topic 2, MySQL HeatWave	23
Topic 3, MySQL HeatWave Security	4
Topic 4, MySQL HeatWave Backup and Recovery	6
Topic 5, MySQL HeatWave Monitoring and Troubleshooting	2
Topic 6, MySQL HeatWave Migration	12
Total	76

QUESTION NO: 1

You want to keep an OCI MySQL DB system as the source database and continuously copy its transactions to an external MySQL instance for disaster-recovery purposes.

Which replication feature should you configure?

A. Inbound replication

B. Outbound replication

C. A read replica

D. A HeatWave cluster

ANSWER: B

Explanation:

Outbound replication copies transactions from a MySQL DB system to an external MySQL instance. In this topology, the OCI DB system is the replication source and the external server receives and applies the source binary-log events.

Inbound replication has the opposite direction: an external MySQL instance is the source and the MySQL DB system is the target. Read replicas scale read workloads from a source DB system, and a HeatWave cluster accelerates analytical query processing rather than replicating transactions to an external server.

QUESTION NO: 2

Automatic backup retention of a DB system is disabled. Which operation deletes all automatic backups of the DB system?

A. Enabling point-in-time recovery

B. Disabling delete protection

C. Deleting the DB system

D. Disabling point-in-time recovery

ANSWER: C

Explanation:

Deleting the DB system deletes all of its automatic backups when automatic backup retention is disabled. In MySQL HeatWave on OCI, automatic backups are normally retained according to the configured retention period, including after the DB system itself is deleted. However, disabling automatic backup retention changes that lifecycle behavior: the automatic backups are not preserved independently after deletion of the associated DB system. Consequently, deleting the DB system removes the DB system and all automatic backups belonging to it.

This distinction is important when decommissioning an environment. If recovery capability or historical backup retention is required after the DB system is removed, create and retain a manual backup before deleting the DB system. Manual backups have a separate lifecycle and can be retained independently, subject to the applicable service and tenancy policies. Oracle documents the relationship between automatic-backup retention and DB system deletion in its MySQL HeatWave backup documentation. See Backup and Restore for MySQL HeatWave and Deleting a DB System.

QUESTION NO: 3

Which two are true about MySQL HeatWave configuration resources? (Choose two.)

A. You can set a value for any global MySQL variable in a configuration.

B. You must copy the configuration before changing any variable values.

C. You can view custom configuration sets with the OCI CLI oci mysql db-system get command.

D. You can compare configurations in the same compartment in OCI console.

ANSWER: B D

Explanation:

“You must copy the configuration before changing any variable values.” is correct because a MySQL HeatWave configuration is treated as a reusable configuration resource. To alter the variable values represented by an existing configuration, you create a copy and make the desired changes in that new configuration. This preserves the original configuration as a stable, reusable definition for DB systems that may already rely on it, while allowing controlled configuration variations for a different workload or deployment. “You can compare configurations in the same compartment in OCI console.” is also correct. The OCI Console provides a comparison capability for configuration resources, enabling an administrator to inspect differences in MySQL variable settings between configurations located in one compartment. This is useful when validating a copied configuration before associating it with a DB system, or when determining which settings differ between workload-specific configuration sets. Comparing configurations helps make changes auditable and reduces the risk of unintentionally deploying an unsuitable variable set. Oracle documents the configuration resource and its lifecycle in the MySQL HeatWave Service documentation, and the OCI CLI reference identifies configuration as its own MySQL resource type: Oracle MySQL HeatWave Service: Configurations and OCI CLI MySQL Configuration commands.

QUESTION NO: 4

Which sequence of steps must you perform to change the configuration of a high-availability DB system?

A. Disable high availability, edit the DB system, change the configuration, and enable high availability.

B. Stop the DB system, edit the DB system, change the configuration, and start the DB system.

C. Edit and change the configuration of all secondary instances, switch over to a secondary instance, and edit and change the configuration of the former primary instance.

D. Edit the DB system and change the configuration.

ANSWER: B

Explanation:

The required sequence is: stop the DB system, edit the DB system, change the configuration, and start the DB system. For a high-availability MySQL HeatWave DB system, stopping the service first places the DB system in the required state for a configuration change that cannot be applied while it is running. You then open the DB system’s edit workflow and select or modify the associated MySQL configuration as needed. Saving the edit applies the requested configuration association or configuration setting change to the DB system definition. Starting the DB system afterward brings the high-availability DB system back online with the selected configuration in effect across its managed HA topology. This sequence ensures that the configuration is applied through the supported OCI DB system lifecycle rather than attempting to modify individual HA members independently. Oracle documents DB system configuration management and the process for editing DB systems in the MySQL HeatWave Service documentation: Configuring a DB System and Managing DB Systems.

QUESTION NO: 5

A MySQL DB system has an endpoint in an OCI VCN subnet. The VCN is not configured with VCN peering, VPN access, and FastConnect. You want to connect to the MySQL DB system directly from an OCI Cloud Shell session.

Which two are true? (Choose two.)

A. You must attach the Cloud Shell session to a VCN subnet that can connect to the MySQL DB system.

B. You must install mysql client programs in the Cloud Shell session.

C. The MySQL DB system subnet must be in the Cloud Shell tenancy home region.

D. The VCN subnet of the Cloud Shell session must have a Service Gateway in its route table.

E. The VCN subnet of the Cloud Shell session must have an Internet Gateway in its route table.

ANSWER: A C

Explanation:

You must attach the Cloud Shell session to a VCN subnet that can connect to the MySQL DB system. Cloud Shell normally runs outside the customer VCN, so it does not inherently have a private network path to a DB system endpoint in a VCN. Cloud Shell private networking lets a session be attached to a subnet. That subnet must have network connectivity to the DB system subnet, including appropriate route rules and security rules permitting the MySQL connection. When there is no peering, VPN, or FastConnect path, attaching Cloud Shell to a subnet in the applicable VCN provides the required private access path.

The MySQL DB system subnet must be in the Cloud Shell tenancy home region. Cloud Shell private-network attachment is available only for VCNs in the tenancy's home region. Therefore, a Cloud Shell session can be attached for direct private access only when the relevant VCN subnet—and consequently the DB system endpoint being reached through that attachment—is in that home region. This regional requirement is specifically important for DB systems that use private IP addressing. Oracle documents Cloud Shell private networking and its home-region limitation at Oracle Cloud Shell documentation. MySQL HeatWave connectivity guidance is available at Connecting to a DB System.

QUESTION NO: 6

Which placement is NOT valid for a high-availability DB system? (Note: AD refers to availability domain and FD refers to fault domain.)

A. Primary instance: AD1-FD1; Secondary instance 1: AD2-FD2; Secondary instance 2: AD3-FD3

B. Primary instance: AD1-FD3; Secondary instance 1: AD2-FD2; Secondary instance 2: AD3-FD3

C. Primary instance: AD1-FD1; Secondary instance 1: AD1-FD2; Secondary instance 2: AD1-FD3

D. Primary instance: AD1-FD3; Secondary instance 1: AD1-FD2; Secondary instance 2: AD1-FD3

ANSWER: D

Explanation:

Primary instance: AD1-FD3; Secondary instance 1: AD1-FD2; Secondary instance 2: AD1-FD3 is not a valid placement because the primary instance and one secondary instance are assigned to the identical availability-domain and fault-domain combination, AD1-FD3. A MySQL HeatWave high-availability DB system consists of three MySQL instances, with one primary and two secondary instances. OCI placement must ensure that the instances are separated so a fault affecting an individual fault domain does not simultaneously affect more than one member of the HA group. When all instances are in one availability domain, they must occupy distinct fault domains; fault-domain identifiers are meaningful within their availability domain. This placement duplicates a fault domain for two HA members and therefore violates the required fault isolation. Proper separation enables MySQL HeatWave to retain redundant members and support automatic failover when infrastructure affecting a single instance or fault domain fails. Oracle documents that HA DB systems use a group of three instances and distribute them across fault domains or availability domains according to the selected topology. See the Oracle MySQL HeatWave High Availability documentation and the OCI fault domain documentation.

QUESTION NO: 7

Which two are true about data loaded into a MySQL HeatWave cluster? (Choose two.)

A. Data is replicated across cluster nodes for high availability.

B. Data is partitioned across cluster nodes for scalability.

C. Data is stored locally in each cluster node's solid state disks (SSD) to increase I/O performance.

D. Data is stored in hybrid columnar format to speed up analytic queries.

E. Data rows are sorted on the primary key to speed up range search.

ANSWER: B D

Explanation:

Data loaded into a MySQL HeatWave cluster is partitioned across the nodes in the cluster. HeatWave automatically distributes table data into partitions, enabling nodes to process portions of a query in parallel. This massively parallel processing design allows analytic workloads to scale as cluster capacity is increased, because CPU and memory resources from multiple HeatWave nodes can participate in query execution. The distribution is managed by HeatWave rather than requiring application-level data sharding.

HeatWave also stores loaded data in an in-memory hybrid columnar format optimized for analytics. This representation combines characteristics useful for efficient scans, compression, and vectorized processing, reducing the amount of data that must be processed for analytical queries. The optimized in-memory columnar representation is separate from the row-oriented storage used by MySQL InnoDB, while HeatWave remains integrated with MySQL for loading and querying data. Together, distributed partitioning and the hybrid columnar format are core reasons HeatWave can accelerate large-scale analytic processing. Oracle documents these architecture and data-loading characteristics in the MySQL HeatWave overview and the HeatWave cluster documentation.