

1. Data Management Introduction

1.1 Oracle Data Management Strategy

1.1.1 Introduction: The Modern App Development Context

- App development paradigms are undergoing rapid transformation.
- Modern app development simplifies and accelerates application deployment, data models, and analytics.
- Oracle Data Management embraces this transformation, offering a platform described as "**simply complete**" and "**completely simple**."

1.1.2 A Simply Complete Platform for Development

1.1.2.1 The Converged Database

- Provides best-of-breed support for all data models and workloads.
- Eliminates data fragmentation by supporting multiple data types within a single database.
- Enables unique queries and transactions across any data type.
- Supported data models include:
 - JSON
 - Graph
 - Text
 - Relational
 - Blockchain
 - Spatial

1.1.2.2 Support for All Modern Workloads

- Converged data management supports all transactional and analytical workloads.
- Allows running any combination of workloads on any combination of data.
- This includes:
 - Key-value
 - IoT (Internet of Things)
 - Operational
 - Data Warehouse
 - Data Lake
 - Machine Learning

1.1.2.3 Simplified Decentralized App Development

- Oracle's decentralized database architecture simplifies deployment and operation.
- Supports modern development techniques:
 - Coding events
 - Data events

- API-driven development
- Low-code
- Geo-distribution
- Autonomous Database (ADB) now supports the MongoDB API.

1.1.2.4 Autonomous Database (ADB) for Productivity

- Features a set of automated tools for management, provisioning, tuning, and patching.
- Simplifies complex database engineering with:
 - Auto-indexing
 - Auto-partitioning
 - Elasticity (automatic scaling up/down based on workload)
- Provides self-service tools for analytics and data access.
- Allows developers to focus on solving business problems rather than database management.

1.1.3 A Completely Simple Platform for Running Applications

1.1.3.1 Mission-Critical Operational Support

- Oracle provides proven solutions for high availability, security, and operations.
- Mission-critical capabilities are built into the Oracle Data Management architecture.
- Simplifies development by eliminating the need for complex sharding and data protection.

1.1.3.2 Integrated Data Protection and Security

- The Oracle Autonomous Database includes:
 - Disaster recovery
 - Replication
 - Backups
 - Security
- Provides automated, transparent solutions for minimizing risk, managing complexity, and ensuring availability.

1.1.4 Conclusion: The Big Picture Strategy

- Oracle's Data Management strategy is "**simply complete and completely simple.**"
- It is built on three pillars:
 - The Converged Database
 - Data Management Tools
 - A Best-in-Class Platform
- The platform is focused on enabling modern app development across all data types, workloads, and development styles.
- It is consistently scalable, available, and secure across the entire environment.
- It is the simplest to use due to its available tools and ability to run mission-critical applications.

1.2 Oracle Database Offerings

1.2.1 Deployment Models: Managed vs. Co-managed vs. On-Premise

- **Autonomous Database (ADB):** A fully managed service where Oracle manages both the infrastructure and the database.
- **Co-managed Service (e.g., Database Cloud Services):** Oracle manages the infrastructure, but the customer manages the database.
 - Used for databases not yet on 19c or for packaged applications like E-Business Suite.
- **On-Premise:** The customer manages everything, including infrastructure and the database.

1.2.2 The Business Case for Autonomous Database (ADB)

- ADB offers a significantly lower Total Cost of Ownership (TCO).
- Automation of tedious DBA tasks (best practices, configurations) reduces labor costs.
- **Wikibon Research Analysis:** Running Oracle Database on Amazon RDS was found to be as expensive or 50% more expensive than on-premise, especially with Disaster Recovery.
- In contrast, Autonomous Transaction Processing (ATP) costs were almost half that of on-premise.

1.2.3 Migrating to Autonomous Database

- **Migration Assessment:** A tool is available to analyze an on-premise database and flag any unsupported features for ADB (e.g., pre-19c versions, tables in system schema).
- **Migration Process:** The standard method uses Data Pump to export data from on-premise to the cloud object store, then import it into ADB.
- **Database Migration Service:** A newer, automated service that handles the entire migration process from an on-premise or other cloud source to ADB.
- **Tuning:** Existing tuning and indexes from on-premise databases can be preserved in ADB.

1.2.4 Application and Tool Certification

- Packaged applications like JD Edwards, PeopleSoft, and Siebel are certified to run on ADB (E-Business Suite certification is upcoming).
- A fully managed service is available to move the entire application stack (middle tier and database) to the cloud.
- Third-party applications (e.g., MESTEC, MineSense, NEC, Zebra) are also being certified for ADB.
- Popular analytic tools (e.g., Tableau, BusinessObjects) are certified to run against ADB.

1.2.5 Customer Success and Use Cases

- **TaylorMade:** Successfully moved a large on-premise data warehouse to Autonomous Data Warehouse (ADW), achieving lower TCO and better performance.
- **Wilson Truck Lines:** Rebuilt an application in two days using APEX and ADB that had originally taken three months to build on Amazon with traditional coding, demonstrating a 45x productivity increase.

1.2.6 Advanced Capabilities: APEX (Application Express)

- APEX is Oracle's low-code tool for rapidly building data-driven applications.
- It enables development estimated to be 10x to 50x faster than traditional coding.
- Features include:
 - Out-of-the-box tools for forms, reports, and faceted search.
 - Easy UI customization (colors, themes).
 - Automated middle-tier management (connection, state, data type mapping).
- Millions of APEX applications exist, with thousands built daily.

1.2.7 The Analytic Platform: Beyond Data Warehousing

- ADB is more than a data warehouse; it is a converged analytic platform.
- It includes self-service tools for data loading, ETL transformations, and building OLAP models.
- Built-in analytic technologies include:
 - Graph Analytics
 - Spatial Analytics
 - Machine Learning (including AutoML for automated model generation)
- Supports data lake architectures and can query data in the database and object stores (on Oracle Cloud and other clouds).
- **Seattle Sounders / English Premier League:** Use ADW to build powerful analytic environments and ML models for sports statistics.

1.2.8 Business Programs and Incentives

- **Bring Your Own License (BYOL):** Allows existing Oracle customers to use their on-premise licenses on Oracle Cloud Infrastructure (OCI).
- **Cloud Lift Service:** A free, global engineering team to help customers migrate on-premise databases to ADB.
- **Support Rewards Program:** Customers receive a \$0.25-\$0.33 reward for every dollar spent on OCI, which can be applied to their technology support bills.

1.2.9 Summary of ADB Benefits

- **Fully Managed Service:** Includes Exadata infrastructure, database software, and provides best-practice configurations for security, availability, and performance.
- **Automation:** Automatically patches and tunes the database.
- **Cost Reduction:** Lowers TCO via automation, higher productivity, and less downtime.
- **Ease of Modernization:** A straightforward path to modernize existing Oracle Databases (19c) to the cloud.
- **Free Tier:** An "Always Free Autonomous Database Service" and "Live Labs" (developer.oracle.com/livelabs) are available for hands-on testing and learning.

1.3 Multicloud and Hybrid Cloud

1.3.1 The Current Enterprise Cloud Landscape

- Less than 20% of mission-critical enterprise workloads have been migrated to the cloud.
- Nearly 70% of enterprises are currently moving or planning to move on-premise workloads to the cloud.
- A common dilemma: Organizations have chosen a strategic cloud provider but still have mission-critical systems (like Oracle Engineered Systems) on-premise with no clear migration path.

1.3.2 Defining a Multicloud Strategy

- **Definition:** The use of multiple cloud computing and storage services in a single, heterogeneous architecture.
- **Goal:** To distribute assets across several cloud environments to eliminate reliance on any single provider.
- **Driver:** 81% of organizations work with two or more public cloud providers, recognizing that no single provider can meet all needs.

1.3.3 Key Benefits of a Multicloud Approach

- **Best-of-Breed Innovation:** Select the best cloud for each specific workload.
- **Reduced Risk & Increased Redundancy:** Deploying applications across multiple providers enhances resilience.
- **Reduced Latency:** Choose local cloud vendors based on facility locations to improve performance.
- **Compliance & Governance:** Adhere to data sovereignty laws and regulations by using multiple storage providers.
- **Economics:** Avoid the cost of building and maintaining private data centers by using public cloud infrastructure.
- **Reduced Vendor Lock-in:** Spreading investment across multiple providers mitigates risk and increases negotiating power.

1.3.4 Connecting Clouds: Performance, Reliability, and Security

- Private connectivity, avoiding the public internet, is required for predictable performance and security.
- **Oracle Cloud Infrastructure FastConnect:** Provides a direct, private connection between the customer and Oracle.
- **Third-Party Connectivity (e.g., Megaport Cloud Routers - MCR):** Enables multicloud connectivity, such as between an OCI Virtual Cloud Network and an Amazon Virtual Private Cloud (VPC).
- These connections offer redundant circuits and leverage multiple cloud providers' economics.

1.3.5 Oracle's Hybrid and Multicloud Solutions

- **Global Public Regions:** Over 30 public regions with a consistent, automated architecture for global scale and disaster recovery.

- **Dedicated Region:** A full-scale Oracle Cloud region installed and operated within a customer's own data center, managed by Oracle, to meet data sovereignty and low-latency requirements.
- **Cloud@Customer:** Offers high-performance Exadata and Autonomous Database solutions on the customer's premises, delivered and managed as a service by Oracle.
- **Partnership with Microsoft Azure:**
 - **Interconnected Clouds:** Direct, low-latency (<2ms) connection between Oracle Cloud and Microsoft Azure via FastConnect and ExpressRoute.
 - **Unified Identity:** Single sign-on and automated user provisioning for managing resources across both clouds.
 - **Collaborative Support:** Enables running workloads across clouds (e.g., Oracle applications on Azure with an Oracle Database backend).

2. Converged Database

2.1 How to Simplify Application Development?

2.1.1 The Challenge of Modern Application Development

- Modern enterprise applications must be data-driven, anticipating customer needs and creating value from data in real-time.
- These applications operate on diverse data types (spatial, documents, sensor, transactional) pulled from multiple sources.
- They create value through advanced analytics like:
 - Machine Learning (e.g., for real-time recommendations, fraud detection)
 - Graph Analytics (e.g., to identify influencers in a community)
 - Spatial Data (e.g., to track deliveries)
- Development uses modern paradigms (microservices, event processing, API-driven, low-code) and requires continuous delivery with zero downtime.

2.1.2 The Pitfalls of Single-Purpose Databases

- Initially, single-purpose databases seem beneficial as they offer a convenient model for a new project.
- However, this approach leads to significant problems:
 - **Data Fragmentation:** Data is siloed across multiple databases.

- **Increased Complexity:** Mid-project changes often require combining data models (e.g., running analytics on a document store).
- **Steep Learning Curve:** Each database has its own proprietary APIs and languages, moving away from standard SQL.
- **Vendor Lock-in:** Proprietary APIs can lock an application to a specific database or cloud provider.
- **Complex Application Code:** Requires writing and maintaining code to propagate and sync data between different stores.
- **Operational Overhead:** Each database requires separate tuning, securing, scaling, troubleshooting, and patching, demanding specialized skills.

2.1.3 The Oracle Solution: The Converged Database

- Oracle's solution is a converged database that eliminates data fragmentation by providing native support for all modern data types and models within a single database.
- Key principles of the converged database:
 - **Native Support:** Built-in support for all data types (e.g., Graph, JSON, Key Value) in one product.
 - **Engineered to Work Together:** Different data types are engineered to work synergistically.
 - **Example:** A single query can span multiple models (find a customer's friends (Graph) who watched similar movies (JSON) and gave a five-star rating (Key Value)).
 - **Uniform API:** Provides easy access for developers using standard SQL and REST for all data types and workloads.
- This approach does not mean a return to a monolithic architecture. Applications can still be modularized, with each service using a logically separate data container (a Pluggable Database or PDB).

2.1.4 Simplifying Development Paradigms

- The converged database pairs each modern development methodology with a synergistic data technology to eliminate complexity.
- **Example - Microservices:** Oracle simplifies microservices by allowing each service to store its data in an isolated Pluggable Database (PDB), providing the necessary independence and agility within a unified database architecture.

2.1.5 Summary of Benefits

- Provides synergistic data technologies for modern development.
- One converged database for all data types and models, engineered to work together.
- Greatly simplifies both development and operations.

2.2 Oracle Autonomous JSON Database

2.2.1 Core Architecture: Bridging NoSQL and SQL

- A JSON "collection" is a logical abstraction over a database table.

- Creating a collection (e.g., "employees") automatically creates a table with a single JSON column to store the documents.
- Data is stored in OSON, Oracle's optimized binary JSON format.

2.2.2 Document Operations and Their SQL Translation

- **Inserting Data:** The `insertOne` function inserts a JSON document as a row in the backing table via a SQL INSERT statement.
- **Querying Data:** The `find` method uses a filter expression (Query by Example) to select documents.
 - This operation is translated into a SQL SELECT query on the backing table.
- **API Access:** The API provides simple `put`, `get`, and `update` operations for JSON objects.

2.2.3 Dual-Mode Access: NoSQL and SQL

- **For Application Developers:** The document API offers a simple, non-verbose way to persist data, lowering the barrier to entry.
- **For Data Analysts/Scientists:** The backing table remains fully accessible using standard SQL.
- The database kernel has extensions to natively and efficiently process JSON data via SQL.

2.2.4 Ecosystem and Tooling Compatibility

- **MongoDB Compatibility:** Developers familiar with MongoDB can use their existing skills, drivers, and development tools.
- **SQL Tooling:** The database works with the vast ecosystem of existing SQL-based tools (e.g., Power BI, Tableau) that can connect to Oracle databases.
- **Relational Views:** Relational views can be created over JSON collections, allowing traditional BI tools to access the data seamlessly.

2.2.5 Key Benefit

- Oracle Autonomous JSON Database provides the combined advantages of a document database and a full-featured relational database in a single, converged platform.

2.3 Developing on Oracle Autonomous Database - Using Graph

2.3.1 Graph Database Fundamentals

- Graph databases explicitly store relationships between data entities, facilitating the discovery and understanding of connections.
- **Vertices:** Represent entities (e.g., bank accounts).
- **Edges:** Represent the connections or activities between entities (e.g., cash transfers), which can have properties (e.g., amount, date) and a direction.

2.3.2 Business Applications of Graph Analytics

- **Financial Services:**

- Detecting fraud and money laundering by identifying unusual patterns like "distribution and accumulation" and cycles (money returning to its origin account through multiple layers).
- **Manufacturing & Network Management:**
 - Performing impact analysis (e.g., the effect of a component design change on assemblies, or a network router failure on connected systems) using path analysis.
- **Recommendation Systems:**
 - Identifying customer clusters for product recommendations.
- **Flexible Data Modeling:**
 - The flexible schema accommodates products with varying or new attributes easily.

2.3.3 Oracle Graph Features and Capabilities

- **Availability:** Free and included in all Oracle Database editions.
- **Core Components:**
 - A means to model and store graph data.
 - The **Property Graph Query Language (PGQL)** for querying, which extends SQL to specify graph patterns (e.g., cycles).
 - Over 60 built-in, parallel, in-memory graph algorithms for analytics.
 - Visualization tools to interact with results.
- **Types of Graph Algorithms:**
 - **Global:** Operate on the entire graph (e.g., community detection, ranking).
 - **Local:** Operate on portions of the graph (e.g., shortest path finding).

2.3.4 Customer Use Case: Fraud Detection

- A customer enhanced their fraud detection by combining cycle detection with the **PageRank** algorithm.
- **Insight:** Using PageRank (which considers the importance of connected vertices) to identify significant accounts within cycles of 6-7 hops proved more effective than just using the number of transactions (in/out-degree).
- **Result:** This approach led to a significant reduction in false positives.

2.4 Developing on Oracle Autonomous Database - Using Spatial

2.4.1 The Value and Ubiquity of Spatial Data

- Spatial data identifies real-world locations, relating disparate entities (people, events, activities).
- It helps identify patterns, determine relationships, and understand correlations to answer business questions (e.g., customer demographics, infrastructure planning).
- Virtually all enterprise data has a location component (assets, incidents, transactions).

2.4.2 Oracle's Spatial Solution: An Integrated Approach

- Location is managed as an integrated attribute within the converged database.

- The solution supports all types of geospatial data, from store locations and street networks to satellite imagery and 3D city models.
- It consists of three core parts:
 1. **Core Database Functions:** Hundreds of spatial operators and functions within the database for querying and analysis via SQL.
 2. **Developer APIs & Services:** Components for developers using Java, Python, JavaScript, and RESTful services to build applications (e.g., for visualization, route calculation).
 3. **Spatial Studio:** A self-service, no-code tool for analysts.

2.4.3 Spatial Studio: No-Code Access and Capabilities

- **Purpose:** An intuitive, browser-based tool for non-technical users to create maps and perform spatial analysis without coding.
- **Key Features:**
 - Provides no-code access to the database's spatial functions.
 - Renders results in interactive maps.
 - Performs **geocoding** (converting addresses to coordinates) using the Oracle Maps Cloud Service.
 - Assists developers with data preparation, index creation, and building complex analytic workflows.
- **Integration:** Results can be handed off to Oracle Analytics Cloud for further analysis.

2.4.4 Deployment and Availability

- Spatial functionality is included with Autonomous Database at no additional license cost but requires compute resources.
- Spatial Studio is available as a deployable JEE application and on the Oracle Cloud Marketplace.

2.4.5 Example Business Questions Addressed

- Which is the nearest warehouse with specific items in stock?
- Are customer complaints clustered in a specific area?
- Do service areas of retail branches overlap?
- Which assets are located in a flood zone?
- Does a planned route cross restricted areas?

2.4.6 Summary

- The Oracle Autonomous Database provides native functionality to store, query, and analyze spatial data using SQL or various development frameworks.
- Spatial Studio offers no-code access to this powerful functionality.
- Spatial features can be used in conjunction with other database capabilities like JSON, Graph, and ORDS (Oracle REST Data Services).

3. Exadata and Base Database Service

3.1 Oracle Base Database Service Overview - Part 1

3.1.1 Service Definition and Core Concept

- A service to deploy full-featured Oracle Databases on Virtual Machines (DB Systems) in Oracle Cloud Infrastructure (OCI) regions.
- It is a **co-managed service**: Oracle manages the cloud infrastructure, and the customer manages the database and its VM contents.
- Provides root access to the customer's database VM.

3.1.2 Key Features and Supported Configurations

- **Database Editions & Versions:** Supports Standard Edition and Enterprise Edition workloads for versions 12c, 19c, 21c, and 23ai.
- **Deployment Types:**
 - Single Instance Virtual Machine DB System
 - 2-node RAC (Real Application Clusters) Virtual Machine DB System
- **Security:** Features a multi-level security model with always-on database encryption (Transparent Data Encryption).

3.1.3 Licensing Models

- **License Included:**
 - A subscription model that includes the Oracle Database license and all Enterprise Edition options and packs.
 - Ideal for new applications or for using features beyond current on-premise licenses.
 - Cost-efficient with elastic OCPU metering.
- **Bring Your Own License (BYOL):**
 - Designed to minimize costs when migrating existing workloads.
 - Customers can use their existing licenses in the cloud.
 - Grants rights to use key Enterprise Edition options (TDE, Diagnostic Pack, Tuning Pack, etc.) without additional licenses.
 - Eligible for the **Support Rewards Program**, providing a credit against on-premise support bills.

3.1.4 License Included Tiers

- **Standard Edition 2:** Includes Multitenant (up to 3 PDBs), Machine Learning, Spatial and Graph.
- **Enterprise Edition:** Adds Data Guard and Enterprise Management Packs (Data Masking, Tuning, Diagnostics).
- **Enterprise Edition High Performance:** Adds Lifecycle/Cloud Management Packs, Partitioning, Advanced Compression, and Advanced Security. Required for Multitenant with more than 3 PDBs.
- **Enterprise Edition Extreme Performance:** Includes all features plus Active Data Guard, RAC, and Database In-Memory. Required for RAC deployments.

3.1.5 Cloud Automation and Lifecycle Management

- Provides a rich set of automation functions for database lifecycle tasks, reducing admin workload and configuration errors.
- Automates:
 - Database provisioning
 - Resource scaling
 - Patching
 - Backup and recovery
 - High Availability and Disaster Recovery (with Data Guard)

3.1.6 Resource Configuration and Scaling

- **Compute (OCPU):** Billed by the second. Customers select the number of OCPUs and corresponding memory.
- **Storage:** Uses OCI Block Volumes. Customers specify the amount of storage for data (up to 80 TB) and recovery (up to 20 TB).
 - I/O performance increases as allocated storage is scaled up.
- **Backups:** Choice of Object Storage or the default Recovery Service, which offers more features and better Recovery Time/Point Objectives (RTO/RPO) at the same cost.
- **Scaling Limitations:**
 - Cannot scale a single instance VM to a 2-node RAC VM.
 - Scaling a 2-node RAC system is done in a rolling manner (one node at a time).

3.1.7 Compute Shapes (VM Types)

- **Ampere Flexible Shape:**
 - Most economical. Supports 1 to 57 OCPUs. Memory is 8 GB per OCPU.
 - Only supports single-node systems and Logical Volume Manager (LVM). Does not support Standard Edition.
- **Intel Flexible Shape:**
 - Supports 1 to 32 OCPUs. Memory is 16 GB per OCPU.
 - Supports single instance and 2-node RAC, Standard and Enterprise Edition.
- **AMD Flexible Shape:**
 - Supports 1 to 64 OCPUs. Memory is 16 GB per OCPU.
 - Supports single instance and 2-node RAC, Standard and Enterprise Edition.
- **Standard Fixed Shapes (Legacy):**
 - 1 to 24 OCPUs. Memory is 15 GB per OCPU.
 - Scaling requires changing to a different shape size.

3.2 Oracle Base Database Service Overview - Part 2

3.2.1 Storage Architecture Options

- **Logical Volume Manager (LVM):**
 - Available for single-node deployments.
 - Used for fast provisioning.
- **Oracle Automatic Storage Management (ASM):**
 - The default storage management architecture.
 - Required for 2-node RAC deployments.
 - Automatically creates DATA (80%) and RECO (20%) disk groups.
 - Underlying block storage provides triple mirroring for data protection.
- **Storage Management:** Cloning is supported for both LVM and ASM.
- **Storage Scaling:** Storage size can be specified at launch and scaled up online after creation.

3.2.2 Oracle Cloud Infrastructure High-Availability Concepts

- **Region:** A localized geographical area containing independent OCI infrastructure.
- **Availability Domain (AD):** One or more fault-tolerant data centers within a region, interconnected by a low-latency network.
- **Fault Domain (FD):** A grouping of hardware within an Availability Domain. Each AD contains three FDs to isolate against hardware failures.

3.2.3 High Availability and Disaster Recovery Configurations

- **2-Node RAC Deployment:**
 - Instances are automatically deployed on separate physical servers in different Fault Domains within the same Availability Domain.
 - This provides protection against server, network, and power failures.
- **Data Guard:**
 - Can be deployed in a different Availability Domain for high availability (HA).
 - Can be deployed in a different region for disaster recovery (DR).
- **Built-in Best Practices:** Cloud automation ensures optimal configuration for performance, availability, and security upon deployment.

3.2.4 Maximum Availability Architecture (MAA) and Licensing

- The service supports all Oracle MAA technologies.
- Feature availability is tied to the licensing tier:
 - **Flashback & Backup/Recovery:** Available in all editions.
 - **Multitenant:** Available in all editions (up to 3 PDBs without a license). The PDB limit is lifted (up to 4,098) with High Performance or Extreme Performance tiers.
 - **Data Guard:** Available in all Enterprise Edition tiers.
 - **Active Data Guard & RAC:** Require the Enterprise Edition Extreme Performance tier.
 - **Application Continuity:** Requires the Extreme Performance tier (as it depends on Active Data Guard and/or RAC).

3.2.5 Defense-in-Depth Security Strategy

Security is implemented at multiple layers across the entire stack:

- **Database Layer Security:**
 - Transparent Data Encryption (TDE)
 - Data Redaction, Masking, and Subsetting
 - Key Vault
 - Database Vault
 - Database Firewall
 - Data Safe
- **Operating System Layer:**
 - Hardened Oracle Linux 7 image.
 - Minimal software packages, disabled unnecessary services, and secure configuration parameters.
- **Network Layer:**
 - Isolation via OCI Virtual Cloud Networks (VCNs) and VLANs.
 - Oracle Native Network Encryption for client-to-database connections.
- **Service & Compliance:**
 - The service undergoes external audits (PCI, HIPAA, ISO 27001).
 - Includes additional features like antivirus scanning, automated alerting, and vulnerability scans.

3.3 Exadata Database Service Overview

3.3.1 Service Definition and Core Value Proposition

- A service to run Oracle Database on the Exadata platform in the Oracle Cloud.
- Delivers the highest performance, availability, and scalability for Oracle Database.
- Provides a complete Oracle Database installation with all Enterprise Edition features, options, and management packs included (e.g., RAC, Database In-Memory, Multitenant).

3.3.2 Exadata Platform Foundation

- Built on the proven Exadata Database Machine architecture.
- Features a fault-tolerant design with:
 - Scale-out database servers and intelligent storage servers.
 - A fully redundant, high-performance RDMA over Converged Ethernet (RoCE) network fabric.
- Includes key Exadata innovations: SQL Offload, Smart Flash Cache, Storage Index, and Hybrid Columnar Compression.

3.3.3 Deployment and Management Model

- **Co-managed Service:** Oracle manages the underlying infrastructure (hardware, firmware, networking), while the customer manages the database and has root access to the Database VM.
- **Provisioning:** Uses web-based wizards for quick system and database deployment.
- **Service Isolation:** Dedicated infrastructure with no overprovisioning, ensuring predictable performance.

3.3.4 Economics and Licensing

- **Subscription Model:** Pay a monthly fee based on the Exadata shape and the number of enabled OCPUs. No upfront capital cost.
- **Licensing Models:**
 - **License Included:** Subscription includes all database licenses.
 - **Bring Your Own License (BYOL):** Use existing on-premise licenses.
- **Elastic Scaling:** Pay only for allocated OCPUs and scale them up/down elastically.

3.3.5 Key Differentiators and Benefits

- **100% Compatibility:** Fully compatible with on-premise Oracle Databases and applications, enabling easy migration.
- **Complete Cloud Automation:** Provides customer-controlled automation for lifecycle tasks (provisioning, patching, backups) via UI and APIs.
- **Deployment Flexibility:** Available in multiple locations:
 - **Exadata Cloud Infrastructure (Public Cloud):** In Oracle's public cloud data centers.
 - **Exadata Cloud@Customer:** The full Exadata service deployed in the customer's own data center.
 - **Oracle Database@Azure:** Native offering in Microsoft Azure.
- **Cloud@Customer Specifics:**
 - Data remains in the customer's data center, meeting data residency and security requirements.
 - Offers the same cloud automation, APIs, and subscription economics as the public cloud.
 - Managed by Oracle, with the control plane in a public OCI region.

3.3.6 Comparison to Autonomous Database

- **Exadata Database Service:** Co-managed; customer controls the database and OS, ideal for full customization.
- **Autonomous Database:** Fully managed; self-driving, self-securing, self-repairing; ideal for lowest TCO and freeing DBAs from management tasks.

3.4 Cloud Management Responsibilities

3.4.1 Shared Responsibility Model

- **Oracle's Responsibilities (Manages the Infrastructure):**

- Physical database servers, storage servers, and internal RoCE network fabric.
- Hypervisor software.
- Infrastructure patching, security scans, and updates (performed within customer-scheduled maintenance windows).
- Oracle staff are **not authorized** to access customer guest VMs.
- **Customer's Responsibilities (Manages the Database & VM):**
 - Everything running **above the hypervisor** in the database Virtual Machine (VM).
 - This includes: database software, grid infrastructure, application data, schemas, and encryption keys.
 - Installing and managing additional software within the guest VM.
 - Controlling user access to the guest VM.

3.4.2 Customer Management via Cloud Automation

- Customers manage their responsibilities using provided Cloud Automation tools (UI and REST APIs).
- These tools provide control over operational and database lifecycle tasks, including:
 - VM and database creation and deletion.
 - Patching.
 - Backup and recovery.
 - Resource scaling (up and down).
 - High Availability and Disaster Recovery (HA/DR) deployments with Data Guard.
- Benefits: Reduces admin workload, avoids configuration errors, and frees DBAs to focus on value-added tasks.

3.4.3 Exadata Cloud Infrastructure Architecture

- The architecture consists of database servers and storage servers connected by a secure, high-performance RoCE network.
- **Database Servers:** Host one or more Database VMs, which contain the Grid Infrastructure, database software, and cloud automation tools.
- **Networking:**
 - **Internal RoCE Network:** Provides ultra-high-speed communication between VMs and storage.
 - **Client/Backup Networks (50 Gbps):** Connects to the customer's Virtual Cloud Network (VCN) for application connectivity, backups, and Data Guard.

3.4.4 Exadata Cloud@Customer Architecture

- **Purpose:** Deploys the full Exadata Database Service within the customer's own data center for those who cannot use the public cloud due to data residency, latency, or compliance requirements.
- **Core Components:**
 - A full Exadata system deployed on-premise.

- Local control plane servers that establish a **secure tunnel** to the Oracle Cloud Infrastructure (OCI) control plane in a public cloud region.
- **Management Model:**
 - Identical to the public cloud service: Oracle manages the infrastructure, and the customer manages the database VMs.
 - The customer uses the same OCI console/APIs to manage databases, but all data remains on-premise.
 - End-users and applications connect to it like any other on-premise database server.
- **Oracle Operations Access:** The Cloud Operations team can access the infrastructure layer for issue resolution, with full auditing and access controls provided to the customer.

3.5 Billing and Licensing

3.5.1 Licensing Models

- **License Included:**
 - A subscription that includes licenses for Oracle Database Enterprise Edition, all Enterprise Manager Packs, and all database options.
 - Ideal for customers without existing licenses or those wanting to use features beyond their current entitlements.
- **Bring Your Own License (BYOL):**
 - Designed to minimize costs by using existing Oracle Database Enterprise Edition and option licenses.
 - Grants rights to use key database options without separate on-premise licenses, including:
 - Transparent Data Encryption (TDE)
 - Data Safe
 - Diagnostic Pack
 - Tuning Pack
 - Data Masking and Subsetting Pack
 - Real Application Testing
 - Includes the Exadata storage server software license.
 - **Note:** Oracle Standard Edition is not supported on Exadata Database Service.

3.5.2 Cost Management via Elastic Scaling

- Licensing costs are based on the number of OCPUs in use.
- A key cloud advantage is the ability to **vertically scale OCPUs online** with no service disruption.
- This allows customers to:
 - Scale up OCPUs to handle workload spikes.
 - Scale down OCPUs when demand drops.

- Contrasts with on-premise systems, which must be sized for peak capacity, leading to underutilization and higher fixed costs.

3.5.3 Defense-in-Depth Security

Security is implemented across multiple layers:

- **Database Layer:**
 - Transparent Data Encryption (TDE)
 - Data Redaction, Masking, and Subsetting
 - Key Vault
 - Database Vault
 - Database Firewall
 - Data Safe
- **Compute Layer (Exadata):**
 - Secure, compliant technical implementation guidelines.
 - Minimal software packages to reduce security risk.
 - Token-based SSH access for service accounts.
- **Network Layer:**
 - Isolation via OCI Virtual Cloud Networks (VCNs) or VLANs.
 - Oracle Native Network Encryption for client connections.
 - Dedicated, isolated networks for client and backup traffic.
- **Compliance:** The service undergoes stringent audits (FedRAMP, HIPAA, PCI).

3.5.4 Operator Access Control (For Exadata Cloud@Customer)

- A critical feature for regulated industries that require control and visibility over infrastructure access.
- Customers can control Oracle Operations staff access to their on-premise Exadata Cloud@Customer infrastructure by:
 - Granting access to specific components for a limited time with specific privileges.
 - Observing and recording all operator commands and keystrokes in real-time.
 - Terminating an operator's session immediately if necessary.
- This provides the confidence for regulated customers to leverage cloud benefits while maintaining strict control.

3.6 Exadata Database Lifecycle Management - Administering Exadata

3.6.1 Custom Software Images

- **Purpose:** Create standardized "gold images" for Grid Infrastructure and Database software to enforce organizational standards.

- **Capabilities:** Custom images can include specific versions, Release Updates (RUs), bundle patches, and one-off patches.
- **Process:**
 - Created as a resource within the tenancy before provisioning or patching.
 - Stored in Oracle-managed object storage.
 - Can be created via Console or API by selecting the image type (Database or Grid Infrastructure), release, and desired patches.

3.6.2 Database Home Creation

- A Database Home is created on an Exadata VM Cluster.
- During creation, you select a database image—either an Oracle-provided image or a custom software image created previously.

3.6.3 Database Creation

- **Process:** Initiated from the Exadata VM Cluster details page.
- **Key Configuration Steps:**
 - Specify database name, version, and an optional PDB name.
 - Select a Database Home (existing or new).
 - Set administrator credentials.
 - Configure backup destination and schedule.
 - Choose encryption: **Oracle-managed keys** or **customer-managed keys** (using OCI Vault).
 - **For Cloud@Customer:** Integrate with on-premise **Oracle Key Vault (OKV)** for centralized, customer-controlled key management.

3.6.4 Pluggable Database (PDB) Management

- A Container Database (CDB) with one PDB is created by default.
- **Lifecycle Operations (via Console/API):**
 - **Create:** Add new PDBs to an existing CDB.
 - **Clone:** Create a copy of a PDB.
 - **Local Clone:** Within the same CDB.
 - **Remote Clone:** In a different CDB (can be across VM clusters).
 - **Refreshable Clone:** A remote clone that can be refreshed with source data.
 - **Relocate:** Move a PDB to a different DB system.
 - **Start/Stop:** Control the state of a PDB.
 - **Restore:** Restore a PDB from a backup to the latest point or a specific time.
 - **Delete:** Remove a PDB.

3.6.5 Data Guard Configuration

- **Purpose:** Provides real-time disaster recovery and high availability.
- **Setup:** Simplified to a few clicks or a single API call using cloud automation.

- **Configuration Steps:**
 - Select the standby VM cluster (in another Availability Domain or Region).
 - Choose Data Guard type (Standard or **Active Data Guard**).
 - Select or create a Database Home for the standby.
 - Provide a unique database name and matching admin password.
- **Role Management Operations:**
 - **Switchover:** Planned role reversal (primary and standby swap roles).
 - **Failover:** Unplanned role reversal during a primary database failure.
 - **Reinstate:** Return a failed former primary database to service as a standby.

3.6.6 Data Guard Prerequisites

- Primary and standby systems must be in the same OCI compartment.
- Database versions must be identical.
- Each database must have a unique DB_UNIQUE_NAME.
- **Networking:**
 - Same Region: DB systems must use the same Virtual Cloud Network (VCN).
 - Different Regions: The VCNs in each region must be peered.
 - Security rules must allow stateful TCP traffic on the SCAN listener port (default 1521).

3.7 Exadata Database Lifecycle Management: User Managed Maintenance Tasks - Part 1

3.7.1 Maintenance Responsibility Matrix

- **Oracle-Managed Infrastructure Maintenance:**
 - Applies quarterly updates to all underlying hardware and firmware.
 - Includes: physical compute servers, storage servers, internal network switches (RoCE fabric), PDUs, ILOM, and control plane components.
- **User-Managed Maintenance (Required Quarterly, not to exceed 180 days):**
 - Patching the Oracle Grid Infrastructure software.
 - Patching the Oracle Database software on the Database VMs.
 - Updating the operating system on the Database VMs.

3.7.2 Prerequisites and Best Practices for Patching

- **IAM Access:** Users must have the appropriate Identity and Access Management (IAM) policies to perform maintenance operations.
- **Pre-Patch Best Practices:**
 - **Backup:** Always backup the database before applying any updates.

- **New Database Home:** Consider moving the database to a new Database Home instead of updating the existing one.
- **Patch Order:** Update the **Grid Infrastructure first**, as its patch level determines the highest database version the cluster can support.
- **Precheck:** Run the precheck operation before the maintenance window to identify and resolve potential issues.
- **System Health:** Ensure all servers and database instances are running.
- **Free Space:** Ensure /u01 (GI Home) and /u02 (DB Home) each have at least 15 GB of free space.
- **Use OCI Interfaces:** Use OCI management interfaces (Console/API) for update operations.

3.7.3 Identifying Current Patch Levels and Available Updates

- **Location:** The **VM Cluster Details** page shows current patch levels for:
 - Exadata Image (includes OS)
 - Grid Infrastructure
 - Database Homes
- **Available Updates:** The **Updates** page lists available patches, organized into three sections:
 - **Exadata VM Cluster OS:** Contains OS updates.
 - **Grid Infrastructure:** Shows GI patches or upgrades.
 - **Database Home:** Lists standard and custom database software images.
- **Update History:** The **Update History** page provides a log of all patching operations (successful or failed) performed via the OCI console/API. It does not show updates applied via command-line tools like `dbaascli`.

3.7.4 Updating the Guest VM OS (Exadata Image)

- **Process:** Automated via the OCI Console or API.
- **Features:**
 - Includes a **precheck** function to validate system readiness.
 - Provides a **rollback** option in case of issues.
- **Availability:** Only the four latest Exadata image minor versions are available for update.
- **Procedure:**
 - From the **Updates** page, select the desired OS update.
 - Use the **Actions** menu to either:
 - **Run Precheck** (recommended first step).
 - **Apply** the update (this runs a precheck automatically before applying).

3.7.5 Patching the Grid Infrastructure (GI)

- **Process:** Also performed via the **Updates** page or REST API.
- **Procedure:**

1. Select the desired GI patch.
 2. Use the **Actions** menu to **Run Precheck** or **Apply** the patch.
- **Key Consideration:** The GI update is performed in a **rolling fashion** (one node at a time). The database instance on the node being updated will be unavailable during its update cycle.

3.8 Exadata Database Lifecycle Management: User Managed Maintenance Tasks - Part 2

3.8.1 Updating the Oracle Database Home

- **Two Patching Paths:**
 1. **Update an Existing Database Home:** Applies the patch to all databases using that Home.
 2. **Move Database to a New Database Home:** Creates a new Home with the desired patch level and moves individual databases to it. This is the **preferred method** as it is faster, less disruptive, and only affects a single database.
- **Update Process (via OCI Console/API):**
 1. From the **Updates** page, select the desired Database Home patch (from a standard or custom image).
 2. Use the **Actions** menu to **Run Precheck** (recommended) or **Apply** the patch (which runs a precheck first).
 3. Database Home updates are applied in a **rolling manner** across RAC instances.

3.8.2 Using Custom Database Software Images for Updates

- **Two Paths:**
 1. **Create a New DB Home with Custom Image & Move Database:** Patches a single database.
 2. **Update the Existing DB Home with Custom Image:** Patches all databases in that Home simultaneously.

3.8.3 Moving a Database to a New Database Home

- **Procedure:**
 1. From the **Database Details** page, select **Move to Another Database Home**.
 2. Select the target Database Home with the desired patch level.
 3. Click **Move Database**.
- **Status:** During the move, the database and Database Homes show an "updating" status. A "failed" status indicates an issue, with details provided.

3.8.4 Service Impact During Updates

- **Goal:** Achieve zero database service downtime through **RAC rolling updates**.
- **Impact:** Maximum compute performance is temporarily reduced while RAC instances are restarted.
- **Best Practice:** Follow Exadata Cloud Maximum Availability Architecture (MAA) documentation for continuous application availability.

3.8.5 Upgrading the Grid Infrastructure (GI)

- **Purpose:** Enables provisioning of Database Homes and databases using the most current Oracle Database software version.
- **Process:**
 - Performed via OCI Console or API.
 - Executed in a **rolling fashion** (one node at a time).
 - Database instances on the node being upgraded are unavailable during its cycle.
- **Prerequisites & Restrictions:**
 - Run an upgrade precheck before the maintenance window.
 - Not available if an infrastructure maintenance operation is scheduled within 24 hours.
 - **The following operations are blocked during a GI upgrade:** Enabling Data Guard, switchover/failover, node management (start/stop/reboot), CPU scaling, Database Home/database provisioning, database restore, and editing IORM settings.

3.8.6 Upgrading a Database

- **Pre-Upgrade Preparation:**
 1. **Backup:** Perform a full on-demand backup and disable automatic backups (as they conflict with the upgrade).
 2. **Test:** Test the new version on a non-production system.
 3. **Precheck:** Run a database upgrade precheck.
 4. **Create Target DB Home:** Create a Database Home containing the target database software version.
 5. **Ensure PDBs are Open:** All Pluggable Databases (PDBs) in the container must be openable to avoid upgrade failure.
- **Automatic Upgrade Steps:**
 1. An automatic precheck is run.
 2. A guaranteed restore point is set for flashback in case of failure.
 3. The database is moved to the target Database Home.
 4. The Database Upgrade Assistant (DBUA) performs the upgrade.
- **Procedure:**
 1. On the **Database Details** page, select **More Actions > Upgrade**.
 2. In the **Upgrade Database** screen, select the target Oracle Database version and the target Database Home.
 3. Run the precheck and resolve any issues.
 4. Click **Upgrade Database**.
- **Data Guard Consideration:**
 1. For databases using Data Guard, **upgrade the standby first**.

2. Redo apply is disabled during the upgrade. Verify redo apply and open mode after the upgrade is complete.

4. Autonomous Database & Tools

4.1 Describe Autonomous Database Architecture and Integration

4.1.1 Core Value Proposition and Benefits

- **Reduced Management:** Eliminates complexity of manual database operations (provisioning, securing, patching, backing up, tuning), reducing administration costs by up to 80%.
- **Accelerated Development:** Enables immediate provisioning for new applications, eliminating delays and saving costs.
- **Cloud-Native Data Management:** A self-managing service that automatically optimizes performance, security, and availability for the current workload without pre-planning.

4.1.2 High Availability and Reliability

- **Fault-Tolerant Architecture:** Automatically establishes a triple-mirrored, scale-out cluster within one region.
- **Disaster Recovery:** Offers optional full standby copy in another region via Autonomous Data Guard for protection against regional failures.
- **Automatic Recovery:** Transparently recovers from physical failures at the server or data center level.

4.1.3 Key Automated Features

- **Provisioning:** Rapidly deploys a mission-critical, scale-out RAC cluster on Exadata infrastructure.
- **Security:**
 - Encrypts all data at rest and all client connections.
 - Automatically applies security patches as soon as they are available.
- **Management:**
 - Applies all OS, storage, VM, and database patches.
 - Uses machine learning to detect and diagnose errors.
 - Automatically configures, tunes for workloads, and scales compute resources (enabling true pay-per-use).
- **Performance:** Automatically manages indexes, optimizer statistics, and data formats to achieve the best performance.

4.1.4 Foundation

- Built on the Oracle Exadata platform.

4.2 Describe the different ADB offerings and license types

4.2.1 ECPU Pricing Metric

- **Definition:** ECPU (Elastic CPU) is an abstract, performance-based compute metric, not tied to physical cores.
- **Benefits:**
 - Provides a durable pricing metric independent of underlying hardware.
 - Enables finer granularity system sizing and auto-scaling.
 - Offers reduced storage costs for Autonomous Data Warehouse (ADW).

4.2.2 Licensing Models

- **Pay-As-You-Go (Universal Credit Model):**
 - Includes all database options.
 - No need for existing Oracle licenses.
- **Bring Your Own License (BYOL):**
 - For customers with existing Oracle Database licenses.
 - Reduces cost compared to Pay-As-You-Go.
 - **BYOL Restrictions:**
 - **Standard Edition:** Maximum of 32 ECPUs.
 - **Over 64 ECPUs:** Requires existing Real Application Clusters (RAC) licenses in addition to Enterprise Edition.

4.2.3 Service Level Agreement (SLA) Requirements

- To achieve the 99.995% SLA with the BYOL model, customers must have licenses for:
 - Oracle Enterprise Edition
 - Real Application Clusters (RAC)
 - Active Data Guard

4.2.4 Universal Credit Payment Structures

- **Pay-As-You-Go:** Pay for services used; includes necessary licenses.
- **Annual Universal Credits:** A pre-paid, one-year commitment that provides a discount and can be applied to eligible OCI services in any region.

4.2.5 License Flexibility

- Customers can dynamically switch between BYOL and License Included (Pay-As-You-Go) models without incurring downtime.

4. Autonomous Database & Tools

4.3 Create Autonomous Database Serverless Instances - Auto Scaling

4.3.1 Core Scaling Capabilities

- **Independent Scaling:** Compute (ECPUs) and storage can be scaled up or down independently of each other.
- **Auto Scaling:** When enabled, the database can automatically scale compute up to **three times** the base provisioned ECPU count to handle workload spikes.

- **Availability:** Auto scaling is enabled by default for paid tiers and is not available for the Always Free tier.
- **Zero Downtime:** Scaling operations (manual or automatic) do not require downtime and occur while the database remains open and fully accessible.

4.3.2 Management Process

- Scaling is managed via the **Manage Resource Allocation** option in the OCI Console.
 - The database's lifecycle state shows "**Scaling in progress**" during the operation, then returns to "**Available**" upon completion.
-

4.4 Create Autonomous Database Serverless Instances - Provision ADB

4.4.1 Provisioning Overview

- The process is highly automated, provisioning a scalable, secure, and highly available database on Exadata infrastructure with minimal input.
- The underlying architecture includes:
 - A Database 19c PDB in a CDB.
 - A private, software-defined network.
 - A fully encrypted database on a scalable RAC cluster.
 - Pre-configured automated backups.

4.4.2 Key Configuration Steps

1. **Compartment & Naming:** Select the target compartment and provide a display name and database name.
2. **Workload Type:** Choose the appropriate service (e.g., Transaction Processing, Data Warehouse, JSON Database).
3. **Deployment Type:** Select "**Serverless**".
4. **Compute & Storage:** Set the base number of ECPU's and storage size (in terabytes). **Auto Scaling is enabled by default** for paid tiers.
5. **Admin Credentials:** Set a secure password for the ADMIN user.
6. **Network Access:**
 - Choose between a **public endpoint** (secure access from everywhere) or a **private endpoint** on a subnet.
 - Configure **Access Control Rules** (whitelists) to restrict source IPs/CIDR blocks.
7. **Licensing:** Choose **License Included** (Pay-As-You-Go) or **Bring Your Own License (BYOL)**.

4.4.3 Post-Provisioning Status

- The database enters a "**Provisioning**" state.
- It becomes "**Available**" within minutes, ready for connections and data loading.
- No manual configuration of tablespaces, file systems, or initialization parameters is required.

4. Autonomous Database & Tools

4.5 Create Autonomous Database Serverless Instances - Start and Stop ADB

4.5.1 Manual Start and Stop

- **Purpose:** To conserve resources and pause compute billing. Storage billing continues while the instance is stopped.
- **Process:**
 - Initiated via the **More Actions** dropdown in the OCI Console.
 - **Start:** Changes instance state from **Stopped** -> **Starting** -> **Available**.
 - **Stop:** Changes instance state from **Available** -> **Stopping** -> **Stopped**.
- **Data Persistence:** All data and configurations are retained while the instance is stopped.

4.5.2 Scheduled Start and Stop

- **Purpose:** Automate cost reduction by defining shutdown periods for non-peak hours.
 - **Configuration:**
 - Accessed via **More Actions > AutoStart/Stop Schedule**.
 - Schedule is defined in **Coordinated Universal Time (UTC)**.
 - Can set start times, stop times, or both for each day of the week.
 - **Automation:** Operations can also be scripted using REST APIs.
-

4.6 Describe Autonomous Database Tools

4.6.1 Core Autonomous Database Characteristics

- **Self-Driving:** Automates provisioning, monitoring, tuning, and management, freeing DBAs for higher-value tasks.
- **Self-Securing:** Provides built-in protection against external and internal threats with automated patching and encryption.
- **Self-Repairing:** Automatically prevents downtime, aiming for less than 2.5 minutes of downtime per month, including patching.

4.6.2 Database Actions: The Integrated Tool Suite

A unified web-based interface providing access to a comprehensive set of tools.

4.6.3 Development and Data Management Tools

- **SQL Worksheet:** For running SQL and managing the database.
- **APEX (Application Express):** A low-code development environment for building data-driven applications.
- **REST Data Services:** For creating and managing custom REST APIs for the database.
- **JSON Toolset:** For creating collections, loading, and browsing JSON documents.
- **SQLcl:** A modern command-line interface with auto-complete and support for multiple data formats.

4.6.4 Data and Analytics Tools

- **Data Studio:**
 - **Data Loading:** Tools for importing data.
 - **Data Catalog:** Lists and details database entities.
 - **Data Insights:** Provides automated analysis and search for insights.
 - **Data Sharing:** Configuration for secure data sharing.
- **Machine Learning:** Includes **AutoML** and notebook interfaces for building ML models.
- **Graph Studio:** For creating, querying, and visualizing graph data.
- **Spatial Studio:** For visualizing spatial data on interactive maps.

5. AI Innovations

5.1 Introduction to AI

5.1.1 Core Concepts

- **Artificial General Intelligence (AGI):** The hypothetical ability of a machine to mimic human sensory/motor skills, learning, and intelligence to perform complex tasks autonomously.
- **Artificial Intelligence (AI):** The application of machine intelligence to solve problems with specific, narrow objectives.

5.1.2 The Need for AI

- **Data Overload:** The volume of data exceeds human capacity to interpret and make decisions, requiring AI for speed and effectiveness.
- **Key Drivers:**
 - **Automate Routine Tasks:** Examples include credit approval, insurance claims processing, and product recommendations.
 - **Creative & Cognitive Assistance:** Generating content (stories, code, music) and performing complex cognitive tasks.

5.1.3 AI Domains and Examples

- **Language:** Translation
 - **Vision:** Image classification
 - **Speech:** Text-to-speech
 - **Recommendation Systems:** Cross-selling products
 - **Anomaly Detection:** Fraud detection
 - **Reinforcement Learning:** Self-driving cars
 - **Forecasting:** Weather prediction
 - **Generative AI:** Creating images from text
-

5.2 Select AI

5.2.1 Functionality

- A feature that allows users to query the database using **natural language**.
- Translates natural language questions into SQL queries using **Large Language Models (LLMs)**.
- Can be used in tools like **APEX** and **SQL Developer**.

5.2.2 Key Features and Benefits

- **Ease of Use:** Does not require knowledge of underlying table names or SQL syntax.
- **Pluggability & Security:**
 - Configured via an **AI Profile**.
 - Supports multiple LLM providers (OCI Generative AI, Cohere, OpenAI, Llama).
 - Keeps data secure within the Oracle tenancy; data is not sent to external LLM providers.
- **Future-Proof:** Allows switching to new or fine-tuned models as they become available.

5.2.3 Technical Implementation

- Uses the `SELECT AI SQL` statement.
 - The `DBMS_CLOUD_AI` PL/SQL package is used for configuration.
 - Developers can specify which schemas, tables, or views are accessible to the AI.
-

5.3 Vector Search

5.3.1 Overview

- **AI Vector Search** is built into **Oracle Database 23ai**.
- It enables semantic similarity searches across structured and unstructured data.
- A critical component for **Retrieval-Augmented Generation (RAG)** and Gen AI pipelines.

5.3.2 Core Capabilities

- **Native Vector Support:**
 - **VECTOR Data Type:** For storing vector embeddings.
 - **Vector Generation:** Uses the `VECTOR_EMBEDDING` function with models (via API or loaded ONNX models).
- **Similarity Search:**
 - Uses the `VECTOR_DISTANCE` function to find the closest matches.
 - Supports different distance metrics (e.g., Cosine).
- **Vector Indexes:**
 - Created for performance and to control accuracy via the `TARGET ACCURACY` clause.
 - Uses the `APPROXIMATE` keyword in queries to utilize the index.

5.3.3 Key Differentiators in Oracle Database 23ai

- **Converged Database:** Supports vectors alongside JSON, graph, spatial, and relational data.
- **Similarity Search Over Joins:** Can perform vector searches across normalized enterprise data in a single, efficient SQL query.
- **Enterprise-Grade:** Provides full Gen AI pipeline processing with tight integration into the database.
- **Ecosystem Integration:** Works with third-party frameworks like **LangChain** and **LlamaIndex**.

6. Data Lake, Data Warehouse & ML

6.1 Data Lakehouse on OCI

6.1.1 Concept and Value Proposition

- **The Challenge:** Companies traditionally had to choose between Data Lakes (flexibility for raw data) and Data Warehouses (performance for analytics).
- **The Solution:** The **Oracle Lakehouse** unifies both approaches into a single architecture, eliminating data silos and the need for manual data movement.
- **Key Benefits:**
 - Cost savings and improved agility.
 - Enables reuse and recombination of data for maximum insight.
 - Allows querying of external data lakes (on AWS or Azure) directly via Oracle SQL.
 - Supports a wide choice of tools (OCI Analytics, Tableau, Python, etc.).

6.1.2 The Five Key Elements of the Oracle Lakehouse

1. **Data Warehouse:** For relational/structured data (e.g., **Autonomous Data Warehouse - ADW**).
2. **Data Lake:** For raw/unstructured data, using **Object Storage**.
3. **Managed Open Source Services:** Supports Spark (**OCI Data Flow**), Hadoop, and Redis.
4. **Data Integration:** Tools like **Oracle GoldenGate** and **OCI Data Integration** for moving data.
5. **Data Catalog:** A unified metadata repository for data discovery, governance, and a business glossary.

6.1.3 Core Architecture and Components

- **Data Processing:** A unified architecture where data can be written once and analyzed by any engine (SQL, Spark, ML).
- **Oracle Autonomous Data Warehouse (ADW):** A self-managed, high-performance data warehouse for all data types in a converged database.
- **OCI Object Storage:** An efficient storage layer for landing raw data in its native format.
- **OCI Data Catalog:** The central source of truth for metadata, enabling discovery and governance across all data sources.

- **OCI Data Flow:** A serverless Spark service for processing unstructured data.
- **Analytics & AI Services:**
 - **Oracle Analytics Cloud (OAC):** For predictive analytics.
 - **OCI AI Services:** Pre-built ML models.
 - **OCI Data Science:** For building custom ML models.
- **MySQL HeatWave:** A high-performance, in-memory query accelerator for MySQL.

6.1.4 Summary

- The Oracle Lakehouse is an open, flexible architecture that combines the best of data warehouses and data lakes.
 - It enables writing data once and analyzing it with a variety of tools and services.
 - It accelerates solution development by integrating data, services, and tools across the entire technical landscape.
-

6.2 Oracle Machine Learning Overview

6.2.1 Oracle Machine Learning (OML) Core Concept

- A platform within the **Autonomous Database** that enables data science teams to add machine learning intelligence to applications and dashboards.
- It leverages the power of the database to run ML algorithms directly on the data, **eliminating data movement**.

6.2.2 Key Features and Benefits

- **Performance & Scalability:** In-database algorithms redesigned for high performance.
- **Simplified Architecture:** No separate analytical engines required; supports hybrid and multi-cloud environments.
- **Accessibility for Diverse Users:**
 - **For Data Scientists:** Supports **SQL, Python, and R** interfaces, and allows the use of third-party packages.
 - **For Developers:** Provides **REST APIs** for integration.
 - **For Non-Experts:** Offers **AutoML** via a Python API and a no-code user interface.
- **Automation & Monitoring:**
 - Includes automatic data preparation and integrated text mining.
 - Provides no-code UIs for **data monitoring** and **model monitoring**.
- **Cost:** OML components are **included at no additional cost** with Autonomous Database.

6.2.3 The Machine Learning Workflow (with OML)

1. **Define** the project.
2. **Prepare** the data.
3. **Build** the ML model.
4. **Use** the model (generate predictions, analyze patterns).

5. **Deploy** the solution into applications or dashboards.
(The process is iterative, not strictly linear)

6.2.4 Machine Learning Techniques and Use Cases

- **Classification:** Predicting categories (e.g., customer churn, loan default, high-value customers).
- **Regression:** Predicting numeric values (e.g., forecasting sales, product demand).
- **Clustering:** Segmenting data (e.g., customer segmentation, exploratory analysis).
- **Association Rules:** Identifying relationships (e.g., cross-selling, up-selling opportunities).
- **Anomaly Detection:** Finding outliers (e.g., fraud detection, unusual cases).

6.2.5 Industry Applications

- **Financial Services:** Fraud detection, credit risk, customer retention.
- **Healthcare:** Clinical decision support, disease management, trend analysis.
- **Retail:** Recommendation engines, targeted marketing, supply chain optimization.
- **Energy:** Demand forecasting, predictive maintenance, resource exploration.
- **Transportation:** Route optimization, predictive problem-solving.
- **Government:** Public safety, fraud detection, efficiency analysis.

6.3 Data Mesh Architecture

6.3.1 Core Concept and Mindset Shift

- **Definition:** Data Mesh is a decentralized, domain-oriented architectural and organizational paradigm for enterprise data management.
- **Primary Goal:** Improve business outcomes by treating **data as a product** and focusing on data-centric solutions.
- **Key Mindset Shift:** Requires embracing new practices:
 - **Design Thinking:** For solving complex, cross-functional problems.
 - **Jobs to be Done Theory:** Defines a data product's purpose by focusing on the specific goals of the data consumer.
- **Data Product:** A fundamental building block, defined as a set of observations and complementary code designed to fulfill specific jobs.

6.3.2 The Data Economy and the Challenge of "Dead Data Capital"

- **Three Data Economies Coexist in Firms:**
 1. **Command Economy:** Data from enterprise applications (ERP, CRM) for predefined reports.
 2. **Informal Economy:** Data from websites, mobile apps, used for new value outside core systems.
 3. **Market Economy:** Decentralized data supply and demand (e.g., embedded analytics in internet apps).

- **"Dead Data Capital":** The significant loss of value (estimated at 25% of a firm's value) from data that is unused, trapped in silos, undiscoverable, or ineffectively governed.

6.3.3 Data Strategy and Competitive Advantage

- A firm's data strategy must reinforce its competitive strategy by creating **unique data assets** and using them to enhance differentiation and improve cost position.
- This strategy must balance **data innovation** with **data compliance**, considering legislation, regulation, and ethics.

6.3.4 The Four Principles of Data Mesh Architecture

To translate strategy into architecture, four principles are required:

1. **Data Liquidity:** The ease with which data can flow and be reused.
2. **Data Productivity:** The speed and efficiency of creating value from data.
3. **Data Security:** Protecting data across the decentralized landscape.
4. **Data Governance:** Ensuring data quality, compliance, and trust in a federated model.

6.3.5 Key Architectural Attributes of a Data Mesh

- **Domain-Oriented & Decentralized:** Ownership of data is shifted to business domains.
- **Event-Driven & Streaming-Centric:** Uses a variety of event ledgers (e.g., Kafka, CDC tools, messaging middleware) as a durable event log for the enterprise.
- **Self-Serve Platform:** Provides underlying infrastructure and tools as a platform for domains to build and manage their data products.
- **Federated Computational Governance:** Applies global governance rules through automated, computational means.

6.3.6 What Data Mesh Is NOT

- It is **not** a single cloud data lake with domains and a catalog.
- It is **not** a single vendor product.
- It is **not** an IT consulting project alone.
- It is **not** a Data Fabric (which can support monolithic architectures).
- It is **not** just self-service analytics.

6.3.7 Oracle's Role in Enabling Data Mesh

Oracle provides key solutions for different layers of a Data Mesh architecture:

- **Event-Driven Integration & Streaming:**
 - **Oracle GoldenGate:** For real-time data replication and change data capture (CDC).
 - **Oracle IoT Cloud:** For IoT event services.
 - **Oracle Integration Cloud:** For business activity monitoring and application integration.
- **Data Product Creation & Management:**
 - **Oracle Database (Multi-Model):** For creating curated data sets in various formats (relational, JSON, graph).
 - **Oracle APEX:** For building data-driven, self-service applications.

- **Oracle Data Lakehouse:** For storing operational, raw, and curated data sets.
- **Data Consumption & Analytics:**
 - **Oracle Analytics Cloud:** For visualizations and dashboards.
 - **Oracle Stream Analytics:** For real-time time series analysis.
 - **Oracle Data Science:** For building and deploying machine learning models.

7. Developing on Oracle Database

7.1 Manage Autonomous Database instances: Using REST APIs to manage ADB

7.1.1 Core Concept and Benefits

- **Purpose:** Provides a programmatic alternative to the OCI Console for managing Autonomous Database (ADB) instances.
- **Use Cases:** Enables customized deployment scripts, "gold standard" configurations, and version-controlled infrastructure-as-code.

7.1.2 Technical Implementation

- **API Type:** Standard REST APIs using HTTPS requests and responses (TLS 1.2).
- **Authentication:** All requests must be signed using a private key and the RSA-SHA256 algorithm, based on the HTTP Signatures specification. Requires an SSH key pair in PEM format.
- **Language Support:** Can be called from any language that supports HTTPS (e.g., Python, Java, Node.js, cURL).

7.1.3 Common API Operations

- **Create Database:** Specifies parameters like database name, ECPU count, storage, and license model.
- **Delete Database:** Removes an existing ADB instance.
- **Start/Stop Database:** Controls the runtime state of an instance.
- **Scale Database:** Changes the ECPU count (e.g., scaling to 20 ECPU's).
- **Initiate Backup:** Triggers an on-demand backup.

7.2 Autonomous Database Built-In Tools

7.2.1 The Integrated Tool Suite (Database Actions)

- **Access:** A pre-installed, pre-configured suite of tools launched from the OCI Console via the "Database Actions" button.
- **Goal:** Provides an open, collaborative platform for data loading, preparation, analysis, and application development.

7.2.2 Key Tool Categories and Functions

- **Data Loading & Preparation:**
 - Simplifies loading data from local files or cloud object stores (OCI, AWS S3, Azure Blob).
 - Includes **data profiling** to quickly identify data quality issues (e.g., inconsistent formatting, outliers).
- **Development:**
 - **SQL Worksheet:** For running queries and managing database objects.
 - **APEX (Application Express):** Low-code application development environment.
 - **REST Data Services:** For creating and managing REST APIs on database objects.
- **Data Analysis:**
 - **Analytic Views:** Pre-built semantic models for universal data access.
 - **Data Exploration:** Tools to visually profile and investigate loaded data.
- **Underlying Technology:** Integrates with enterprise tools like **Oracle Data Integrator (ODI)** and **Oracle Analytics Cloud (OAC)**.

7.2.3 Key Benefits

- **Collaboration by Design:** Metadata is defined in a common database layer and catalog, breaking down silos between data engineers, analysts, and scientists.
- **Open Platform:** Supports open standards (SQL, Python) and various data formats (CSV, JSON).

7.3 CI/CD for APEX and Oracle Database Developers - Part 1

7.3.1 CI/CD Core Concept and Value

- **Definition: Continuous Integration (CI) and Continuous Delivery/Deployment (CD)** automate the stages of the application development lifecycle.
- **Key Benefits:**
 - **Consistency & Quality:** Enforces coding standards and security, finds issues faster.
 - **Frequent Releases:** Enables more reliable and secure production deployments.
 - **Solves Development Challenges:** Prevents "cowboy coding," manages concurrent development in a shared database, and provides version control for database and APEX metadata.

7.3.2 Foundational Tools

- **Source Control:** GitHub, GitLab, or Bitbucket.
- **SQLcl:** A lightweight, powerful command-line interface for Oracle Database.
- **Liquibase:** An open-source tool for tracking, managing, and applying database schema changes. It is bundled with SQLcl.
- **Jenkins:** An automation server for creating CI/CD pipelines.

7.3.3 Managing Database and APEX Changes with Liquibase

- **Change Tracking:** Liquibase maintains a changelog table in the database, recording all changes made.
- **Intelligent Script Generation:** Liquibase is schema-aware; it generates differential scripts (e.g., ALTER TABLE) instead of just CREATE statements if objects already exist.
- **APEX Application Export:**
 - Can export an entire APEX application as a single file or split into multiple files for better change visibility.
 - Uses flags (-split, -skipExportDate) to export only meaningful changes, ignoring auto-generated IDs and dates.
- **Data Migration:** Suitable for migrating small, static metadata tables (e.g., lookup tables) via insert statements, but not for large data volumes.

7.3.4 Environment Provisioning with Infrastructure-as-Code

- **Terraform:** Used to automate the creation of cloud infrastructure, including Autonomous Databases.
- **Cost-Effective Testing:** Autonomous Database's pay-per-second model allows for creating temporary, isolated test environments at a very low cost.

7.4 CI/CD for APEX and Oracle Database Developers - Part 2

7.4.1 The CI/CD Pipeline Architecture

- **Core Components:**
 - **Developers:** Commit code to a Git repository (e.g., GitHub).
 - **Repository Webhook:** Triggers the CI/CD pipeline on events like a pull request.
 - **Jenkins Server:** An automation server (often on an OCI Compute VM) that orchestrates the pipeline. It is pre-configured with SQLcl, Git, and Terraform.
 - **OCI Instance Principals:** Allows the Jenkins VM to securely call OCI APIs and create resources (like Autonomous Databases) without storing user credentials.
- **Pipeline Flow:**
 - A developer finishes a feature and creates a **pull request**.
 - This triggers the pipeline, which automatically:
 - **Clones a production-like environment** (using Terraform or other methods).
 - **Applies the new code** (database objects and APEX app) using SQLcl/Liquibase.
 - **Runs unit tests** (e.g., using UTPLSQL).
 - **Checks logs for errors.**
 - If the pipeline passes, the code can be merged into the main branch.
 - The temporary test environment is destroyed.
 - After a sprint, a new version is tagged and deployed through environments (DEV, UAT, PROD).

7.4.2 Creating Isolated Developer Environments

A critical requirement is providing each developer with an isolated, fresh environment that mirrors the main codebase.

- **Primary Methods for Environment Creation:**

- **Autonomous Database (ADB) Clones:** Fast and easy cloning via UI, CLI, API, or Terraform.
- **Database VMs:** Create from a backup (slower).
- **Exadata Cloud Service:** Use sparse cloning for rapid creation.
- **Multitenancy (Pluggable Databases - PDBs):** Create PDB clones via:
 - Database APIs
 - ORDS REST APIs
 - Standard SQL scripts
- **Non-Multitenant Databases (Legacy):**
 - **Guaranteed Restore Points / Flashback Database:** Flash the database back to a clean state after each development cycle.
 - **RMAN Clones:** Using database links.
 - **Data Pump:** Export/import the schema.
 - **ACFS / gDBClone:** Using storage snapshots.

7.4.3 Deployment to Production and Rollback Strategies

- **Deployment:**

- **Manual:** Jenkins creates deployment artifacts (ZIP files); a DBA manually applies them to production. This is common for teams not yet fully trusting automation.
- **Automated:** The pipeline automatically deploys the version-tagged code to production environments. This is reliable because the code has been tested repeatedly in identical, cloned environments.

- **Rollback Strategies:**

- **Multitenant/PDBs:** Rollback the entire PDB to a snapshot.
- **Flashback Database:** Revert the entire database to a guaranteed restore point created pre-deployment.
- **Liquibase Rollback:** Use Liquibase to roll back specific database object changes.
- **APEX Rollback:** Re-install the previous version of the APEX application.
- **Table Flashback:** Flashback specific tables to their pre-deployment state.
- **Roll Forward:** Fix the issue directly in production and then incorporate that fix back into the main code line.

8. Resiliency

8.1 Database Maximum Security Architecture

8.1.1 Foundational Security Principles

- **Objective:** Protect valuable data (intellectual property, financial, personal) from theft and misuse.
- **Core Strategy:** A defense-in-depth approach combining multiple technologies. Omitting any single control creates a vulnerability.
- **Key Control Areas:**
 1. **Assess:** Determine the system's current security state and identify risks.
 2. **Detect:** Identify policy violations and anomalous data access attempts.
 3. **Prevent:** Block unauthorized access through robust control mechanisms.

8.1.2 Common Database Attack Vectors

- **Compromised Accounts:** Targeting administrator or end-user accounts (the most common method).
- **Application Attacks:** Exploiting exposed applications as an entry point.
- **Network Sniffing:** Intercepting unencrypted data in transit on the internal network.
- **Data File/Backup Theft:** Stealing underlying database files, backups, or exports.
- **Unpatched Vulnerabilities:** Exploiting known but unaddressed security flaws.
- **Non-Production Systems:** Attacking less-secure test/development clones of the production database.

8.1.3 Baseline Security Posture: Core Controls

These controls can be implemented using standard database features, often at no extra cost.

- **1. System Assessment & Configuration:**
 - **Goal:** Identify insecure configurations, configuration drift, and missing patches.
 - **Tools:**
 - **Database Security Assessment Tool (DBSAT):** A free, on-premises utility.
 - **OCI Data Safe:** A cloud service (included with Oracle Cloud database services) that also provides assessment.
- **2. Access Control & Authentication:**
 - **Principle of Least Privilege:** Ensure accounts have only the necessary privileges.
 - **Privilege Analysis (Enterprise Edition only):** Identifies and helps remove unused privileges.
 - **Strong Authentication:** Implement multi-factor, Kerberos, or PKI certificate-based authentication.
 - **Centralized Management:** Use Enterprise User Security (Oracle LDAP) or Centrally Managed Users (Microsoft Active Directory).
- **3. Data Protection:**
 - **Encryption:**
 - **Data in Motion:** Use Oracle Native Network Encryption or TLS.
 - **Data at Rest:** Use Transparent Data Encryption (TDE) and protect the encryption keys.

- **Sensitive Data Discovery:** Use DBSAT or Data Safe to locate and classify sensitive data (PII, financial data, etc.).
- **4. Monitoring & Auditing:**
 - **Database Auditing:** Track critical activities such as:
 - Database connections (especially failures).
 - Data Control Language (DCL - user/privilege changes).
 - Data Definition Language (DDL - object creation/modification).
 - **Network Monitoring:** Monitor SQL traffic for suspicious activity.

8.2 Oracle's Maximum Availability Architecture (MAA)

8.2.1 The Problem: Downtime and Data Loss

- **Impact:** Database downtime and data loss have severe financial consequences (e.g., \$350k/hour average, up to \$10M per outage).
- **Prevalence:** 91% of companies experience unplanned data center outages within a two-year period.

8.2.2 Key Terminology and Objectives

- **High Availability (HA):** Uses redundant components (like clusters) to ensure service continuity during hardware failures.
- **Scalability:** Maintains performance as user load increases.
- **Rolling Updates/Patching:** Applies maintenance in a sequential manner to minimize planned downtime.
- **Disaster Recovery (DR):** Protects against a complete site outage with a redundant failover site.
- **Key Metrics:**
 - **Recovery Time Objective (RTO):** The target time to restore service after an outage.
 - **Recovery Point Objective (RPO):** The maximum acceptable amount of data loss measured in time.

8.2.3 The MAA Blueprint and Methodology

- **Definition:** Oracle's Maximum Availability Architecture (MAA) is a best practices blueprint for achieving optimal high availability, data protection, and disaster recovery.
- **Development Method:** Uses **chaos engineering**—proactively testing system resilience by simulating failures (network, server, storage, human error, data corruption) to validate failover and recovery processes.
- **Scope:** Provides reference architectures for different service levels, covering both engineered and non-engineered systems, on-premises and in the cloud.

8.2.4 Core MAA Technologies and Goals

The architecture uses a suite of technologies to achieve four primary goals:

1. **Data Protection:** Minimize data loss using Flashback and Zero Data Loss Recovery Appliance.

2. **Active Replication:** Enable active-active solutions across sites using **Active Data Guard** and **GoldenGate**.
3. **Scale-Out:** Linearly scale compute nodes using **Real Application Clusters (RAC)**, Automatic Storage Management (ASM), and Sharding.
4. **Continuous Availability:** Enable transparent service failover across local or remote sites using Application Continuity and Global Data Services.

8.2.5 Key Technology Deep Dives

- **Oracle RAC (Real Application Clusters):**
 - A clustering technology that eliminates single points of failure by spreading a database across multiple servers.
 - Provides scalability (add nodes without downtime) and protects committed transactions during a node failure.
- **Transparent Application Continuity:**
 - A feature of RAC that protects **in-flight transactions** during an outage.
 - Automatically replays transactions after a failure, making the disruption transparent to the application and end-user.
 - Requires no application changes, only correct driver and connection string configuration.
- **Active Data Guard:**
 - The core solution for disaster recovery.
 - Maintains one or more synchronized, physical replicas (standby databases) of the production database.
 - Standby databases are open for read-only access while being synchronized, enabling reporting offload.
 - Enables fast failover (unplanned) and switchover (planned) to minimize downtime.

8.2.6 MAA in Oracle Cloud Infrastructure (OCI)

- **Oracle Database Cloud Service:** Offers single instance or RAC deployments, automated backups, and cross-region DR with Active Data Guard.
- **Exadata Cloud Service:** Built on Exadata infrastructure for maximum performance and scalability, available as a public cloud service or on-premises via Exadata Cloud@Customer (ExaCC).
- **Autonomous Database:** Built on a highly available, multitenant RAC Exadata infrastructure.
 - **Autonomous Data Guard:** Can be configured with a few clicks for cross-region or cross-availability domain DR.
 - Fully managed by Oracle.

8.2.7 Multicloud Example

- A **Gold MAA** architecture can be implemented in a multicloud environment.
- **Example:** Oracle Database tier (utilizing RAC and Active Data Guard) in OCI, with the application mid-tier running in a third-party cloud (e.g., AWS, Azure).

- Network connectivity is provided by partners like **Megaport** for redundancy.

9. Upgrades and Migrations

9.1 Oracle Database Cloud Migration

9.1.1 Migration Drivers and Service Portfolio

- **Common Reasons for Migration:**
 - Upgrading to a new major database version.
 - Moving to new hardware platforms.
 - Re-platforming to the Cloud.
- **Oracle Cloud Database Services:** Offer a spectrum of options with varying levels of automation and control to meet different SLA requirements.

9.1.2 Migration Strategies

- **With Downtime:** Suitable for applications that can schedule an outage. Opens up simpler migration methods.
- **Zero Downtime / Continuous Operations:** Essential for 24/7 applications. Requires more complex migration methods to maintain availability.

9.1.3 Migration Process and Tools

- **General Process:** Assess -> Choose Product -> Migrate.
 - **Key Tools:**
 - **Cloud Pre-Migration Advisor Tool (CPAT):** Assesses source database compatibility for cloud migration. Now integrated into Data Migration Service (DMS).
 - **OCI Database Migration Service (DMS):** Focuses on ease of use for zero-downtime migrations.
 - **Zero Downtime Migration (ZDM):** Offers more fine-grained control and supports migrations to Exadata Cloud@Customer (ExaCC).
 - **OCI Application Migration:** For moving applications and VMs to the cloud.
 - **GoldenGate Veridata:** Used for validating data consistency between source and target databases.
-

9.2 Database Upgrade Best Practices

9.2.1 Pre-Upgrade Preparation

- **Target Release: Oracle Database 19c** is the recommended Long Term Support (LTS) release, with premier support until April 2027. Avoid Innovation Releases (like 21c) for production.
- **Performance Stability:**
 - **SQL Plan Management (SPM):** Capture and preserve SQL execution plans *before* the upgrade to prevent performance regression.

- **Real Application Testing:** Use Database Replay to simulate production workload on the upgraded system.
- **Statistics:**
 - Gather dictionary and fixed object statistics *before* upgrading (at least 7 days in advance if using AutoUpgrade) to reduce upgrade time.
 - **Fixed Object Stats:** Must be gathered *after* the system is "warmed up" (i.e., running a representative workload), not right after a restart.

9.2.2 The Upgrade Process: Using AutoUpgrade

- **Primary Tool: AutoUpgrade** is the strongly recommended tool for all upgrades.
- **Key Features:**
 - Automates pre-upgrade checks, the upgrade itself, and post-upgrade tasks.
 - Can upgrade single or multiple databases with one command.
 - Provides detailed logs in multiple formats (TXT, HTML, JSON, XML).
 - Automatically fixes common pre-upgrade issues.
- **Process:**
 - Create a simple configuration file specifying source and target details.
 - Run `AutoUpgrade -mode ANALYZE` to perform pre-upgrade checks.
 - Run `AutoUpgrade -mode DEPLOY` to execute the upgrade.

9.2.3 Fallback Strategy and COMPATIBLE Parameter

- **Primary Fallback:** Use **Guaranteed Restore Points** and **Flashback Database**. AutoUpgrade can automate this.
 - **Critical:** This is only possible *before* users connect and enter new data after the upgrade (pre "go-live").
- **COMPATIBLE Parameter:**
 - **When to Change:** Raise the COMPATIBLE parameter to the new version (e.g., 19.0.0) **7-10 days after** the successful upgrade, during a subsequent maintenance window.
 - **Why Wait:** Changing COMPATIBLE prevents using Flashback Database or a downgrade for fallback. The delay provides a stabilization period.
 - **Value:** Always set to the default major release value (e.g., 19.0.0), not a specific Release Update version.

9.2.4 Post-Upgrade Actions

- **Statistics:** Do **not** regather optimizer statistics immediately after the upgrade. Existing statistics are preserved and valid.
- **System Statistics:** Generally avoid regathering system stats post-upgrade, as it often causes more problems than it solves.
- **Parameters:** Minimize non-default parameters and underscores/events. Use the upgrade project as an opportunity to remove unnecessary ones and move closer to default configurations.

10. MySQL and NoSQL

10.1 Describe Heatwave MySQL

10.1.1 Background and Market Position

- MySQL is the #1 open-source database and the second most popular overall.
- Widely used by major companies (Twitter, Facebook, Netflix, Uber) across various industries.
- Popular for its ease of use, reliability, and performance, especially as part of the LAMP stack.

10.1.2 Core Challenge and HeatWave Solution

- **Challenge:** MySQL is optimized for OLTP but not for analytics, forcing users to maintain two separate databases with complex, stale ETL processes, increasing cost and security risk.
- **Solution: MySQL HeatWave** is a massively parallel, in-memory query accelerator integrated directly into the MySQL Database Service.
- **Key Value Proposition:** A **single, unified platform** for both transactional processing (OLTP) and real-time analytics, eliminating the need for a separate analytics database and ETL.

10.1.3 Key Features

- **Real-Time Analytics:** Changes from MySQL transactions are replicated to the HeatWave cluster in real time.
- **Lakehouse Capability (HeatWave Lakehouse):**
 - Query up to 500 TB of data in object storage (e.g., CSV, Parquet files) without loading it into the MySQL database.
 - Data is processed in its object store location, enabling analytics on non-MySQL data.
- **In-Database Machine Learning (HeatWave AutoML):**
 - Build, train, and deploy ML models using SQL within the database.
 - No separate ML service or data movement required.
 - Automates the ML lifecycle (algorithm selection, feature engineering, etc.).
- **MySQL Autopilot:**
 - Uses machine learning to automate database management.
 - Key features: Auto-provisioning, auto-shape prediction, auto data placement, auto query plan improvement, auto thread pooling, and auto error recovery.

10.1.4 Security and Management

- **Fully Managed Service:** OCI handles backups, patching, monitoring, and OS maintenance.
- **Security:** Built on OCI's Gen 2 cloud infrastructure, with data encryption at rest and in motion.
- **Enterprise Edition:** Based on MySQL Enterprise Edition, including advanced security, scalability features, and support from the MySQL team at no extra cost.

10.2 & 10.3 Oracle NoSQL Database Cloud Service Overview

10.2.1 Addressing Modern Application Challenges

- Designed for applications that require:
 - High velocity data ingestion (e.g., IoT, social media).
 - Low, predictable latency (<10ms).
 - Rapid innovation and dynamic data models.
 - "Always-on" availability and on-demand scaling.
 - Support for structured, semi-structured, and unstructured data.

10.2.2 Service Definition and Core Features

- **Serverless & Fully Managed:** Oracle manages all infrastructure (servers, storage, clustering, software, backups).
- **Elastic:** Provision throughput (read/write units) and storage; scales automatically with workload.
- **Flexible Data Models:** Supports and allows interoperability between:
 - **Key-Value**
 - **Document (JSON)**
 - **Columnar (Fixed Schema)**
- **Run Anywhere:** Available as a cloud service, on-premises, or in a hybrid configuration, using the same application code.
- **Tunable Consistency:**
 - **Eventual Consistency:** 1 read unit (may return stale data).
 - **Absolute Consistency:** 2 read units (guarantees the latest data).

10.2.3 Common Use Cases

- Social Media (user-generated content)
- IoT (sensor data ingestion and analysis)
- Customer 360 (aggregating data from multiple sources)
- Online Gaming (player stats, leaderboards)
- User Profile Management
- Product Catalogs

10.2.4 Billing and Performance

- **Pricing Model:** Pay for provisioned **throughput** (Read Units, Write Units) and **storage** (GB) on an hourly basis.
- **Performance:** Built on OCI Gen 2 infrastructure with NVMe storage, providing predictable low latency without over-subscription.

10.2.5 Development and Operations

- **APIs & SDKs:** Drivers for Java, Python, Node.js, Go, and Spring Data.

- **SQL Support:** A rich SQL interface for all data models, including complex queries, aggregations, and JSON path expressions.
- **Secondary Indexes:** Powerful indexing on any field, including nested JSON data.
- **Time to Live (TTL):** Auto-expiration of data at the table or row level.
- **GeoJSON Support:** For spatial data and queries (intersect, inside, within distance).
- **ID Generation:** Auto-incrementing identity columns.
- **Tools:**
 - **Cloud Console:** Web UI for basic operations.
 - **NoSQL Cloud Simulator:** Free local simulator for development and testing without cloud costs.

10.2.6 Key Differentiators

- **Seamless Multi-Model:** Single data store for key-value, document, and columnar data with full interoperability.
- **Tunable ACID Transactions:** Shard-local full ACID compliance and adjustable consistency.
- **True Hybrid Cloud:** Deploy the same application seamlessly across cloud, on-premises, and hybrid environments.