Oracle Database Introduction

What is Oracle Database?

Definition:

- Oracle's flagship product
- Advanced Relational Database Management System (RDBMS)
- Developed by Oracle Corporation
- First commercially available SQL-based database (1979)

Key Characteristics:

- Handles multiple data types
- Manages large amounts of data
- Operates on both traditional and cloud environments
- Uses SQL for data access and management

About Relational Databases

What is a Database?

- Organized collection of information treated as a unit
- Purpose: collect, store, and retrieve related information
- Used by database applications

Evolution:

- First generation: Hierarchical and Network databases
- Modern standard: Relational Model (E.F. Codd, 1970)

Relational Model Components:

- Structures: Well-defined objects to store/access data
- Operations: Actions to manipulate data and structures
- Integrity rules: Govern operations on data and structures

Database Management System (DBMS)

Definition: Software that controls storage, organization, and retrieval of data

Key Elements:

1. **Kernel code:** Manages memory and storage
2. **Repository of metadata:** Data dictionary
3. **Query language:** Enables applications to access data

RDBMS vs Traditional DBMS:

- Distinguishes between logical and physical operations
- Physical operations are transparent to applications
- Implements the relational model

Oracle Database History

Major Milestones:

1977: Software Development Laboratories founded

- Founders: Larry Ellison, Bob Miner, Ed Oates
- Later became Oracle Corporation (1983)

1979: Oracle V2 released

- First commercially available SQL-based RDBMS

1983: Oracle Version 3

- First relational database on mainframes, minicomputers, and PCs
- Written in C for portability

Oracle Database Evolution (Continued)

Version 4: Multiversion read consistency

Version 5 (1985): Client/server computing and distributed databases

Version 6: Enhanced disk I/O, row locking, scalability, backup/recovery

- Introduced PL/SQL language

Oracle7 (1992): PL/SQL stored procedures and triggers

Oracle8 (1997): Object-relational database, partitioning

Oracle8i (1999): Internet protocols, server-side Java support

---

Oracle Database Evolution (Modern Era)

Oracle9i (2001):

- Oracle Real Application Clusters (RAC)
- Oracle XML Database (XML DB)

Oracle Database 10g (2003):

- Grid computing
- Oracle Automatic Storage Management (ASM)
- Self-managing and self-tuning capabilities

Oracle Database 11g (2007):

- Enhanced manageability, diagnosability, and availability
- Simplified information infrastructure

Oracle Database 12c (2013):

- Designed for the Cloud
- Multitenant architecture
- In-Memory column store
- JSON document support

Oracle Database Release Types

- **Long Term Support Release**
  - 5 years of Premier support followed by 3 years of Extended Support
  - Recommended for production databases
  - Currently is 19c (PS ends April 30, 2024)
  - The next one is scheduled to be 23c
- **Innovation Release**
  - 2 year of Premier Support, no Extended Support
  - Recommended for studying the new features in development environments
  - Examples: 12c, 18c, and 21c

Oracle Database Editions

Enterprise Edition:

- For large organizations
- Supports high-volume OLTP and data warehousing
- Advanced analytics, security, and functionality options

Standard Edition:

- Basic Oracle functionality
- Suited for workgroup, department-level, or web applications

Personal Edition:

- Single-use development environments
- Nearly all Enterprise Edition components
- Cost-effective for developers

Express Edition:

- Free entry-level edition
- Simple to install and manage
- Easy upgrade path to advanced products

Schema Objects

What is a Schema?

- Collection of logical data structures (schema objects)
- Owned by database user
- Has same name as the user

Important Schema Objects:

- **Tables:** Most important schema object
- **Indexes:** Second most important schema object
- Other objects: Views, sequences, procedures, etc.

Key Point: Schema objects are user-created structures that directly refer to database data

---

Tables

Definition: Describes an entity (e.g., employees)

Components:

- **Table name:** Identifies the entity
- **Columns:** Attributes of the entity
- **Rows:** Instances of the entity

Column Properties:

- Name
- Data type
- Width

Integrity Constraints:

- Rules for columns (e.g., NOT NULL)
- Force specific values or behaviors

Example: Employees table with columns: employee_id, first_name, last_name

---

Indexes

Purpose:

- Optional data structure on table columns
- Increase performance of data retrieval

Characteristics:

- Logically and physically independent of data
- Can be dropped and created without affecting tables
- No impact on applications when modified

When to Use:

- Applications frequently query specific rows
- Applications query range of rows
- Need to locate requested rows efficiently

Data Access - SQL

Structured Query Language (SQL):

- ANSI standard language for relational databases
- Set-based declarative language
- Nonprocedural: describes what should be done, not how

SQL vs Procedural Languages:

- Procedural (C): Describes HOW things should be done
- SQL: Describes WHAT should be done

SQL Capabilities:

- Query data
- Insert, update, delete rows
- Create, alter, drop objects
- Control database access
- Guarantee consistency and integrity

SQL Statement Example

Basic SQL Structure:

```
SELECT first_name, last_name
FROM employees;
```

Key Features:

- Simple but powerful
- String of SQL text
- Users specify result wanted, not how to derive it

Oracle SQL:

- Implementation of ANSI standard
- Supports numerous extended features beyond standard SQL

PL/SQL and Java

PL/SQL:

- Procedural extension to Oracle SQL
- Integrated with Oracle Database
- Controls flow of SQL programs
- Uses variables and error-handling procedures

Benefits:

- Store application logic in database
- Built-in functionality deployed anywhere
- Server-side programming

Java Support:

- Store Java program units in database
- Java stored procedures published to SQL
- Call PL/SQL from Java and vice versa

Transaction Management

What is a Transaction?

- Logical, atomic unit of work
- Contains one or more SQL statements
- Either all committed or all rolled back

Example: Funds Transfer

1. Decrease savings account
2. Increase checking account
3. Record transaction in journal

All or Nothing Principle: Oracle Database ensures all operations succeed or fail as a unit

---

Why Transactions Matter

Database vs File System:

- File system: Partial updates can leave inconsistent state
- Oracle Database: Moves from one consistent state to another

Transaction Principles:

- Atomic operation
- Succeeds or fails as a whole
- Maintains data consistency

RDBMS Requirement: Must group SQL statements for commit or rollback

---

Data Concurrency

Definition: Simultaneous access of same data by multiple users

Challenge: Multiple users updating same data simultaneously

Without Concurrency Controls:

- Users could change data improperly
- Compromises data integrity
- Destructive interactions possible

Goal: Reduce or eliminate wait time while preventing destructive interactions

---

Oracle Locking Mechanism

What are Locks?

- Mechanism preventing destructive interaction between transactions
- Controls concurrent access to shared resources

Benefits:

- Ensures data integrity
- Allows maximum concurrent access to data

How It Works:

- Oracle Database uses locks automatically
- Prevents improper data changes
- Maintains consistency during concurrent access

Data Consistency

Requirement: Each user must see a consistent view of data

Consistency Includes:

- Visible changes from user's own transactions
- Committed transactions from other users

Problems Prevented:

- Lost update problem
- Uncommitted changes visible to others

Oracle Enforcement: Always enforces statement-level read consistency

---

Read Consistency Levels

Statement-Level Read Consistency:

- Data returned by single query is committed and consistent
- For a single point in time
- When statement was opened

Transaction-Level Read Consistency:

- All queries in transaction see data from same point in time
- Point in time = when transaction began

Oracle Flashback Query:

- Enables explicit specification of point in time
- Advanced data recovery feature

Oracle Database Architecture Overview

Database Server Purpose:

- Reliably manages large amounts of data
- Multiuser environment
- Concurrent access to same data
- Prevents unauthorized access
- Provides failure recovery solutions

Key Concept: Database and instance work together but are distinct entities

---

Database vs Instance

Database:

- Set of files located on disk
- Store data
- Can exist independently of instance

Instance:

- Set of memory structures managing database files
- System Global Area (SGA): shared memory
- Background processes
- Can exist independently of database files

Oracle Database: Term sometimes refers to both instance and database together

---

Instance and Database Components

Client Process:

- Runs application for each user connection
- Associated with server process

Server Process:

- Has private session memory
- Program Global Area (PGA)

Memory Structures:

- SGA: Shared memory area for instance
- PGA: Private memory for server processes

Physical Structure: Database files store actual data on disk

Multitenant Architecture

Container Database (CDB):

- Single physical database
- Contains zero, one, or many pluggable databases

Pluggable Database (PDB):

- Portable collection of schemas and schema objects
- Appears as non-CDB to Oracle Net client

Non-CDB:

- Traditional Oracle database
- Cannot contain PDBs

Key Innovation: Architectural difference from traditional databases

---

Multitenant Architecture - Problems Solved

Traditional Challenges:

- Hundreds/thousands of databases on different platforms
- Multiple physical servers
- Databases using only fraction of hardware capacity
- Wasted hardware and human resources

Solution: Consolidate multiple physical databases into single database on single computer

---

Multitenant Architecture Benefits (Part 1)

Cost Benefits:

- Reduced hardware costs
- More efficient resource utilization
- Consolidated infrastructure

Operational Benefits:

- Easier data and code movement
- Simplified management and monitoring
- Separation of data and code

Administrative Benefits:

- PDB administrator: manages assigned PDBs only
- CDB administrator: manages entire CDB
- Clear separation of duties

Multitenant Architecture Benefits (Part 2)

Manageability Benefits:

- Easier upgrades by unplugging/plugging PDBs
- Simplified testing in development PDBs
- Individual PDB flashback capability
- Performance limits at PDB level (memory, I/O)

Application Management:

- Application containers for master definitions
- Set of PDBs with common application root
- Install, upgrade, manage applications centrally

Sharding Architecture

Purpose: Handle massive distributed databases

Key Features:

- Single logical database across multiple physical databases
- Horizontal partitioning
- Enhanced scalability

Use Cases:

- Very large datasets
- Global distribution requirements
- High-performance needs

Oracle Database Platform Support

Operating Systems:

- Linux
- Microsoft Windows
- Other major operating systems

Benefits of Cross-Platform Support:

- Flexibility in deployment
- Harness strengths of each OS
- Stable and efficient operations
- Easier management for administrators

Compatibility: Seamless integration across diverse IT environments

---

Key Features of Oracle Database

Advanced Analytics:

- Big data support
- Predictive analytics
- Automated systems
- Quick data processing

High Data Availability:

- Data replication
- Backup capabilities
- Server clustering
- Minimal downtime

Scalability:

- Single server deployment
- Massive distributed database support
- Grows with business needs

Security Features

Multi-Layered Security:

- Advanced data encryption (at rest and in transit)
- Robust user authentication
- Comprehensive access controls
- Security risk monitoring

Protection Against:

- Unauthorized access
- Potential breaches
- Threats and vulnerabilities

Integration: Security measures at every layer from storage to application

---

Oracle Real Application Clusters (RAC)

Definition: Multiple instances accessing single database simultaneously

Benefits:

- Fault tolerance
- Scalability
- High availability
- Load balancing
- Failover capabilities

Use Case: Mission-critical applications requiring uninterrupted service

Redundancy: Protects against single point of failure

Backup and Recovery

Importance: Essential for protecting data and maintaining integrity

Oracle Capabilities:

- Real-time data backups
- Swift disaster recovery
- Full backups
- Incremental backups

Business Benefits:

- Maintains trustworthiness
- Minimizes impact of data disruptions
- Ensures business continuity
- Prepares for unforeseen data loss

Structured Query Language (SQL) in Oracle

Role: Powerful language for precise data management

Capabilities:

- Data manipulation
- Data retrieval
- Complex query execution
- High performance and accuracy

Administrator Benefits:

- Efficient database operations
- Robust data handling
- Precise control over data

Oracle Cloud Integration

Oracle Cloud Infrastructure (OCI):

- Supports traditional and modern applications
- AI and machine learning support
- Multicloud, public, hybrid, and dedicated options

Benefits:

- Scalable and secure environment
- Reduced physical hardware overhead
- Focus on growth and innovation
- Dynamic resource allocation

Tools: Application deployment, data analysis, secure storage

Oracle Database Cloud Services

- **User Manager**: client is responsible for updating the OS, installing, upgrading, and patching Oracle database software.
- **Co-Managed:** Oracle database software comes pre-installed in the machines. Oracle provides tools to facilitate taking database backups, patching database software, upgrading database software, and providing disaster recovery.
- **Autonomous**: fully automated database service that uses machine-learning (ML) for automatically tuning, scaling, and patching Oracle databases.

Key Oracle Products - Cloud Applications

Oracle Fusion Cloud Application Suite:

- Software-as-a-Service (SaaS) applications
- Specific business use cases

Components:

- Fusion ERP: Financials, project management, compliance
- SCM: Supply chain and manufacturing
- HCM: Human capital management
- Sales platforms
- Industry-specific add-ons

Oracle E-Business Suite: Simplifies data entry and management with prebuilt templates

---

Key Oracle Products - Hardware

Engineered Systems:

- Built for scale
- x86 and SPARC-based servers
- Java application systems
- Storage and networking solutions

Oracle Exadata:

- Developed specifically for Oracle Database
- Combines database servers, storage, and networking
- Optimized performance
- Engineered system approach

Key Oracle Products - Middleware

Oracle Fusion Middleware:

- Family of tools for enterprise applications
- Data integration
- Business intelligence
- Content management

Oracle Application Express (APEX):

- Low-code web app builder
- Rapid development

Additional Capabilities:

- IoT functionality
- Big data integration
- Content management

Key Oracle Products - Software

PeopleSoft:

- ERP for HR, financial management, campus solutions

Oracle E-Business Suite:

- Order management, logistics, business functions

Oracle Business Intelligence (OBIEE):

- Reporting and analysis

Oracle Enterprise Manager:

- Advanced DevOps data operations
- Large-scale organization management

Trainingcred

Oracle Database Options and Technologies

Requirement	Database Option/Technology
Hardware failure tolerance (active-active)	Real Application Cluster (RAC)
Hardware failure tolerance (active-passive)	RAC One Node
Disaster recovery (DR)	Data Guard
Disaster recovery (DR) with accessible Standby	Active Data Guard
Replication	Oracle GoldenGate (separate product)
Storage Management (HDD failure tolerance)	Automatic Storage Management (ASM)

Oracle Database Use Cases - Business Operations

Business Applications:

- Enterprise Resource Planning (ERP)
- Customer Relationship Management (CRM)
- Human Resources Management (HRM)

Benefits:

- Interactive insights
- Up-to-date information
- Day-to-day operations support
- Real-time data access

Oracle Database Use Cases - Data Management

Data Warehousing:

- Central repository for structured data
- Vast data storage
- Analysis and reporting facilitation

E-Commerce Platforms:

- Product catalog management
- Inventory tracking
- Transaction data handling
- Back-office solutions

Oracle Database Use Cases - Critical Industries

Financial and Insurance Services:

- Secure, encrypted database
- Payment management
- Transaction tracking
- Risk auditing

Government and Nonprofits:

- Secure public records management
- Classified information handling
- Used by Department of Defense
- Public sector applications

Oracle Database Use Cases - Healthcare and OLTP

Healthcare Services:

- Streamlined operations
- Data integrity maintenance
- Workflow optimization
- Predictive insights

Online Transaction Processing (OLTP):

- Large numbers of database transactions
- Real-time execution
- Multiple concurrent users
- E-commerce, banking, telecommunications

Database Components - Tablespaces

Logical Storage:

- Data stored logically in tablespaces
- Flexible structure management

Physical Storage:

- Data files on disk

Administrator Flexibility:

- Manage structure without affecting user access
- Separate logical from physical concerns

Database Architectures

Configuration Options:

Single-Instance Database:

- One instance accessing one database
- Suitable for smaller deployments

RAC Environment:

- Multiple instances accessing single database
- High availability and scalability
- Fault tolerance

Choice Based On:

- Organization needs
- Scale requirements
- Availability requirements

Performance Optimization

Oracle Database Strengths:

- Handles demanding database applications
- High performance under strenuous conditions
- Supports vast data types and volumes

Optimization Features:

- Works in on-premises and cloud settings
- Scales with business growth
- Manages fluctuating workloads
- Maintains performance during expansion

Continuous Updates and Improvements

Oracle's Approach:

- Regular updates
- Security improvements
- Efficiency enhancements
- Proactive maintenance

Benefits:

- Maintained operational efficiency
- Enhanced security posture
- Adaptation to technological advances
- Reduced downtime

Focus: Keeping Oracle Database at forefront of database technology

---

Tools and Strategies for Efficiency

Comprehensive Suite:

- SQL for data management
- PL/SQL for procedural extensions
- Oracle Cloud for scalability
- RAC for high availability

Advanced Tools:

- Optimize physical structures
- Optimize logical structures
- Enhance performance
- Reduce downtime

Result: Streamlined database operations and enhanced productivity

---

Database Administrator Role

Responsibilities:

- Setting up database environment
- Managing database structures
- Optimizing performance
- Ensuring security

Resources:

- Oracle Support
- Structured Query Language (SQL)
- PL/SQL programming
- Advanced Oracle tools

Focus: Smooth database management operations

Client-Server Model

Architecture:

- Database on server
- User interactions through client machines

Benefits:

- Scalable data management
- Flexible deployment
- Multiple user support
- Centralized data control

Support:

- Multiple concurrent connections
- Efficient resource utilization
- Distributed access

Why Oracle Database Matters

Fundamental Pillar:

- More than just a technological tool
- Underpins critical business processes worldwide
- Essential for large organizations

Core Strengths:

- Manages large data volumes
- Provides easy access
- Ensures high availability
- Maintains strong security

Business Impact:

- Crucial resource for enterprises
- Enables data-driven decision making
- Supports business continuity
- Facilitates growth and innovation

Summary: Oracle Database at a Glance

What It Is: Advanced RDBMS by Oracle Corporation, first commercial SQL database

Key Capabilities:

- Handles multiple data types and large volumes
- SQL and PL/SQL support
- Multi-platform deployment
- Cloud and on-premises operation

Major Features:

- Scalability and high performance
- Advanced security and encryption
- High availability with RAC
- Comprehensive backup and recovery

Why It Matters: Essential tool for enterprise data management, supporting critical business operations across industries worldwide

---

Oracle Database Administrator Common Tasks

- Plan the Database Design
- Install Oracle database and Grid Infrastructure Software
- Create Oracle Databases
- Backup the database
- Control database security
- Tune Oracle database performance
- Implement Database Disaster Recovery (DR) plan
- Migrate the on-prem databases to the Cloud
- Patching database and grid infrastructure

Oracle Database Administrator Common Tasks

- Upgrade Oracle databases
- Create and Manage Oracle clusters (RAC)
- Clone databases
- Monitor database resources growth and determine future demands
- Perform database health check
- Oracle database options and licenses
- Diagnosing and troubleshooting Oracle databases

Oracle DBA Demand in the Job Market

- Bull sources:
  - The introduction of Autonomous Databases in 2018
  - High cost of professional DBAs
  - The introduction of alternative solutions like MySQL and PostgreSQL
  - Cloud services provide easy to use tools
- Push sources:
  - Not all the systems can migrate their databases to Autonomous Database
  - Other cloud database services still need DBAs
  - More IT Systems are being deployed
- IT Recruitment websites demonstrate that DBAs are still required