ORACLE SQL / PL-SQL INTERVIEW PREPARATION

Viewer Document • Technical + Banking Domain + Production Experience

A practical interview-preparation handout covering Oracle SQL, PL/SQL, performance tuning, Oracle 12c → 18c migration, FLEXCUBE, OBPM, OBLM, WebLogic, deployment environments, and client-facing responsibilities.

1. Interview Preparation Roadmap

Area	What to Prepare
Oracle SQL	DDL, DML, DCL, TCL, joins, subqueries, analytical functions, views, materialized views
PL/SQL	Procedures, functions, packages, cursors, collections, bulk processing, exceptions, triggers
Performance Tuning	Execution plans, indexes, statistics, AWR/ASH, wait events, SQL tuning
Migration	12c → 18c assessment, Data Pump, validation, compatibility, testing, rollback
Banking	FLEXCUBE, OBPM, OBLM, transaction processing, batch/EOD concepts
WebLogic	Domain, Admin/Managed Servers, JDBC pools, deployment, logs, troubleshooting
Delivery	DEV/SIT/UAT/PRE-PROD/PROD, release, rollback, client communication

2. Oracle SQL Fundamentals

DDL — Data Definition Language

Used to create or modify database structures.

CREATE • ALTER • DROP • TRUNCATE • RENAME

DML — Data Manipulation Language

Used to insert or change table data.

INSERT • UPDATE • DELETE • MERGE

DCL — Data Control Language

Used to control database privileges.

GRANT • REVOKE

TCL — Transaction Control Language

Used to control transaction boundaries.

COMMIT • ROLLBACK • SAVEPOINT

Frequently Asked Questions

- What is the difference between DELETE, TRUNCATE and DROP?
- How does COMMIT affect transaction control?
- What is the difference between a primary key and unique key?
- When would you use MERGE?
- What is the difference between an INNER JOIN and OUTER JOIN?
- How do analytical functions differ from aggregate functions?

3. Views and Materialized Views

Object	Key Interview Points
View	Logical query definition; normally does not store result data; useful for abstraction and security.
Materialized View	Stores query results physically; can be refreshed; useful for reporting and reducing repeated query work.

Prepare to explain refresh modes, refresh timing, complete vs fast refresh, dependencies, and when a materialized view may improve reporting performance.

4. PL/SQL Development

Procedures

Reusable PL/SQL programs commonly used for business operations. Explain IN, OUT and IN OUT parameters, exception handling, transaction boundaries, and reusable database APIs.

Functions

PL/SQL programs that return a value. Be prepared to discuss calling functions from SQL, deterministic behavior, side effects, and performance considerations.

Packages

A package groups related procedures, functions, variables, constants, cursors and types. Explain package specification vs package body, encapsulation, dependency management and maintainability.

Triggers

Automatically executed in response to database events. Prepare row-level vs statement-level triggers, BEFORE/AFTER timing, DML triggers, compound triggers, and trigger-related performance or mutating-table considerations.

Advanced PL/SQL Topics

Explicit/parameterized/REF cursors • Collections • BULK COLLECT • FORALL • SAVE EXCEPTIONS • Dynamic SQL • Records • Exception handling • Autonomous transactions

5. Oracle Performance Tuning

Use a structured approach: identify the symptom → measure → find the bottleneck → validate the root cause → apply a targeted change → compare before/after results.

Area	Interview Focus

Execution Plan	Explain plan, DBMS_XPLAN, access paths, joins, cardinality, cost and plan changes
SQL Tuning	High elapsed time, CPU, logical reads, physical reads, inefficient joins, unnecessary scans
Indexes	Selectivity, composite indexes, function-based indexes, index maintenance and trade-offs
Statistics	DBMS_STATS, stale statistics, histograms and optimizer behavior
Monitoring	VSQL, VSESSION, V$SQL_MONITOR, AWR, ASH, ADDM
Wait Events	Identify where database time is spent rather than tuning from guesses
Concurrency	Blocking sessions, locks, deadlocks and transaction design

Sample Performance-Tuning Questions

- How do you identify the top SQL statements affecting database performance?
- How do you read an execution plan?
- When would you add an index and when can an index hurt performance?
- What is the role of optimizer statistics?
- What is the difference between AWR and ASH?
- How would you investigate a suddenly slow SQL statement?

6. Oracle 12c → 18c Migration

A migration discussion should cover assessment, planning, execution, validation and rollback—not only data movement.

Phase	Key Activities
Assessment	Inventory schemas/objects, dependencies, application compatibility, unsupported/deprecated features
Preparation	Backup, test environment, migration plan, downtime/RTO/RPO expectations
Migration	Data Pump (expdp/impdp), RMAN or selected migration method; migrate schemas and supporting objects
Validation	Object counts, row counts, invalid objects, constraints, indexes, sequences, grants and application tests
Performance	Compare execution plans, response time, resource usage and critical workloads
Go-Live	Deployment checklist, monitoring, business validation and rollback readiness

Migration Interview Questions

- Why would an organization migrate from Oracle 12c to 18c?
- What pre-migration checks would you perform?
- How would you validate that source and target data match?
- How do you handle invalid objects after migration?
- How would you reduce application downtime?
- What is your rollback approach if production validation fails?

7. Banking Domain — FLEXCUBE

Prepare to explain how database development supports core-banking transactions, account/customer processing, batch processing, interfaces, production support and reconciliation.

Key areas: CASA/DDA concepts • deposits/loans concepts • transaction processing • end-of-day/batch processing • PL/SQL components • data fixes • interfaces • production issue analysis • performance tuning.

Interview prompt: Explain one production issue you handled in a banking application, how you identified the database component involved, the root cause, corrective action, validation and preventive action.

8. OBPM — Oracle Banking Payments

Prepare the payment lifecycle from transaction initiation through validation, processing, status handling, integration, error management and reconciliation.

- Payment transaction flow
- Database tables and PL/SQL components
- Interfaces and message processing
- Error handling and retry concepts
- Batch processing and reconciliation
- Production support and RCA

9. OBLM — Oracle Banking Liquidity Management

Prepare liquidity-management concepts and how transactions, account structures, cash concentration/positioning, interfaces and reporting are supported by database components.

- Liquidity and account structures
- Transaction processing
- PL/SQL and database components
- Interfaces and batch processing
- Production troubleshooting
- Data validation and reconciliation

10. WebLogic

Component	What to Explain
Domain	Logical administration boundary for WebLogic resources
Admin Server	Central administration and configuration
Managed Server	Runs deployed applications
Node Manager	Supports server lifecycle management
JDBC Data Source	Provides application database connectivity
Connection Pool	Reuses database connections and controls capacity
JNDI	Provides named resource lookup
Cluster	Supports scalability and availability

Common production issues: JDBC connection-pool exhaustion, database connectivity errors, long-running SQL, timeouts, application exceptions, high CPU, blocked sessions and deployment failures.

11. Multiple Deployment Environments

Environment	Typical Purpose
DEV	Development and developer testing
SIT	System/integration testing
UAT	Business/user acceptance testing
PRE-PROD	Production-like final validation
PROD	Live production workload

Typical promotion flow: DEV → SIT → UAT → PRE-PROD → PROD

Prepare deployment scripts, version control, change approvals, configuration checks, validation scripts, rollback scripts, deployment verification and post-release monitoring.

12. Client-Facing Experience

A strong client-facing answer should show both technical ownership and communication.

- Requirement clarification and technical discussions
- Production incident and RCA discussions
- Status reporting and delivery coordination
- Release and deployment planning
- Working with application, infrastructure and DBA teams
- Explaining technical issues in business-friendly language
- Knowledge transfer and documentation
- Coordinating across geographically distributed teams

13. Sample 2-Minute Introduction

I have extensive experience in Oracle SQL and PL/SQL development, database performance tuning, production support and banking applications. My technical areas include DDL, DML, DCL, TCL, complex SQL, views, materialized views, procedures, functions, packages, triggers, cursors, collections and bulk processing. I have also worked on Oracle database migration activities, including 12c to 18c migration, covering assessment, data movement, validation, compatibility checks, testing and production deployment. From the banking domain, I have experience with areas such as FLEXCUBE, OBPM and OBLM, along with WebLogic-based application environments. I am comfortable working across DEV, SIT, UAT, PRE-PROD and PROD environments, handling deployment support, production issues, root-cause analysis, performance tuning and client communication.

14. Rapid-Fire Final Revision Checklist

- DDL vs DML vs DCL vs TCL
- View vs Materialized View
- Procedure vs Function
- Package specification vs package body
- Row-level vs statement-level trigger
- BULK COLLECT vs FORALL
- Execution plan and DBMS_XPLAN
- V$SQL / AWR / ASH / ADDM
- Index and statistics strategy
- Blocking, locks and deadlocks

- Oracle 12c → 18c migration phases
- Data Pump and migration validation
- FLEXCUBE transaction and EOD concepts
- OBPM payment lifecycle
- OBLM liquidity concepts
- WebLogic Admin vs Managed Server
- JDBC Data Source and connection pool
- DEV → SIT → UAT → PRE-PROD → PROD
- RCA and production support
- Client communication and documentation

15. Final Interview Answer Framework

For project-based questions, answer in this sequence:

1. Situation / Project Context
2. Your Responsibility
3. Technical Approach
4. Tools / Oracle Features Used
5. Issue or Challenge
6. Resolution
7. Validation / Result
8. Preventive or Improvement Action

Tip: Use a real project example whenever possible. Keep answers structured, explain your own contribution clearly, and distinguish what you personally implemented from what the wider team handled.