SAP ABAP

INTERVIEW ROADMAP

Experience-wise preparation guide for Freshers → 1–2 YOE → 3–5 YOE → 6+ YOE

0–1 YOE Fundamentals & Core ABAP	**1–2 YOE** Development, Debugging & Enhancements
3–5 YOE Advanced ABAP, HANA, RAP & S/4HANA	**6+ YOE** Architecture, Design & Real-world Scenarios

How to use this roadmap: Prepare the fundamentals first, then move toward scenario-based questions as your experience increases. Interviewers generally expect deeper ownership, troubleshooting ability and design thinking with higher experience.

Important: The exact questions vary by project, SAP version, role and company. Use the topics below as a structured preparation checklist—not as a guarantee of exact interview questions.

0–1 YEAR | FRESHER

Core concepts, syntax, database basics and ability to explain what your code does.

1. ABAP Fundamentals

- What is SAP ABAP and where is it used in an SAP system?
- Explain the difference between executable programs, function groups, classes and interfaces.
- What are the different types of ABAP programs?
- What are selection screens, parameters and select-options?
- Explain ABAP events such as INITIALIZATION, AT SELECTION-SCREEN and START-OF-SELECTION.

2. Data Dictionary (DDIC)

- What is a Domain and what is a Data Element?
- Difference between transparent, pooled and cluster tables (know the S/4HANA relevance of these concepts).
- What are primary keys, foreign keys and search helps?
- What are structures, table types and database tables?
- What are check tables and value tables?

3. Internal Tables & Work Areas

- What is an internal table?
- Difference between STANDARD, SORTED and HASHED internal tables.
- What is a work area?
- READ TABLE, LOOP, MODIFY, INSERT, DELETE—when would you use each?
- How do you improve internal-table lookup performance?

4. Open SQL

- Write a SELECT statement with a WHERE condition.
- INNER JOIN vs LEFT OUTER JOIN.
- Why should you avoid SELECT inside LOOP?
- What is SELECT SINGLE and what should you consider before using it?
- How do you handle sy-subrc and sy-dbcnt?

5. Modularization & OOP Basics

- Subroutine vs Function Module vs Method.
- What is a class, object, constructor and inheritance?
- Interface vs abstract class.
- What is encapsulation and why is it useful?

6. Debugging & Basic Tools

- How do you debug an ABAP program?
- What are breakpoints and watchpoints?

- What is ST22 and how do you investigate a short dump?
- What are SE11, SE38/SA38, SE80 and SE37 used for?

1–2 YEARS | JUNIOR ABAP DEVELOPER

Interviewers increasingly test practical development, debugging, enhancements and performance.

1. Performance Optimization

- Why is SELECT inside LOOP considered problematic?
- How do indexes influence database access?
- When would you use JOIN vs FOR ALL ENTRIES?
- What precautions are required with FOR ALL ENTRIES?
- How do ST05 SQL Trace and SAT/SE30 help identify bottlenecks?
- How would you optimize an ABAP report that suddenly becomes slow in production?

2. Enhancements

- What is the Enhancement Framework?
- User Exit vs Customer Exit vs BAdI.
- What is an implicit enhancement?
- How do you identify an enhancement spot for a business requirement?
- Describe a real enhancement you implemented and why you chose that approach.

3. ALV & Reports

- How do you build an ALV report?
- Classic ALV vs SALV.
- How do you add sorting, filtering, totals and custom actions?
- How would you troubleshoot incorrect ALV output?

4. Integration Basics

- What is a BAPI?
- BAPI vs RFC vs Function Module.
- What is IDoc and when is it used?
- What is OData and why is it relevant to SAP UI technologies?
- How would you debug a failed integration?

5. Background Jobs & Production Support

- How do you schedule and monitor a background job?
- How do you investigate a job failure?
- What information do you collect before changing production code?
- Explain a production issue you solved end-to-end.

6. Modern ABAP Introduction

- What are CDS Views?
- What is the difference between a CDS View and a traditional database view?

- What is S/4HANA and how does it affect ABAP development?
- What is Clean Core at a high level?
- What is RAP and why was it introduced?

3–5 YEARS | EXPERIENCED ABAP DEVELOPER

Expect deeper technical discussions, architecture within your scope, performance analysis and scenario-based questions.

1. Advanced ABAP & OOP

- Explain an ABAP OO design you implemented in a real project.
- How do you apply SOLID principles in ABAP?
- When would you use interfaces, composition or inheritance?
- How do you design reusable and testable ABAP code?
- What is ABAP Unit and how do you use it?

2. CDS & Code Pushdown

- Explain CDS Views, associations, annotations and parameters.
- Association vs JOIN—when would you use each?
- What are analytical, consumption and interface views?
- What is code pushdown and why is it important on HANA?
- How would you troubleshoot a slow CDS-based application?

3. AMDP & HANA

- What is AMDP and when is it appropriate?
- AMDP vs Open SQL vs CDS.
- What are the advantages and trade-offs of pushing logic to HANA?
- How do you analyze database-heavy logic?

4. RAP

- Explain the architecture of SAP RAP.
- Managed vs unmanaged implementation.
- Behavior Definition vs Behavior Implementation.
- What are validations, determinations and actions?
- Explain draft handling and locking.
- What are service definitions and service bindings?
- How does RAP support transactional applications and Fiori Elements?

5. OData & Fiori Elements

- Explain OData services and CRUD operations.
- How do you expose a business object?
- What is the role of annotations in Fiori Elements?
- How do you troubleshoot an OData/RAP service issue?

6. Clean Core & Extensibility

- What does Clean Core mean in S/4HANA?
- In-app vs developer vs side-by-side extensibility.
- How would you decide whether to enhance standard SAP or build a separate extension?
- How do you make an extension upgrade-safe?

7. Scenario Questions

- A report is taking 20 minutes in production. Walk through your investigation.
- A business user says a RAP save operation fails only for some records. How do you debug it?
- A standard SAP process needs custom logic. How do you decide the extension point?
- Tell me about a difficult production issue, your diagnosis, fix and preventive action.

6+ YEARS | SENIOR / LEAD

The focus shifts toward technical ownership, solution design, architecture, trade-offs, governance and leadership.

1. Solution & Technical Architecture

- How do you convert a business requirement into a technical design?
- How do you define boundaries between standard SAP, in-app extensions and side-by-side extensions?
- How do you evaluate technical trade-offs such as performance, maintainability and upgrade impact?
- How do you review a technical design proposed by another developer?

2. RAP & S/4HANA Architecture

- How would you design a scalable RAP-based application?
- How do you choose managed vs unmanaged RAP?
- How do you handle draft, locking, authorization and late numbering?
- How do you structure reusable CDS and behavior artifacts?
- How do you keep a RAP solution aligned with Clean Core principles?

3. Performance & Reliability

- How would you investigate a system-wide performance complaint involving ABAP and HANA?
- How do you distinguish an application bottleneck from a database bottleneck?
- What coding and data-access standards do you enforce across a team?
- How do you design for large data volumes and predictable response times?

4. Integration & APIs

- How would you design an API for an S/4HANA business process?
- Compare synchronous and asynchronous integration approaches.
- How do you handle errors, retries, idempotency and monitoring?
- How do you approach security and authorization for exposed services?

5. Production & Incident Management

- How do you lead investigation of a critical production incident?
- How do you balance a quick mitigation with a long-term fix?
- What information should be captured in an incident RCA?
- How do you prevent the same defect from recurring?

6. Leadership & Behavioral Questions

- How do you mentor junior developers?
- How do you conduct code reviews?
- Describe a technical disagreement and how you resolved it.
- How do you estimate and divide a complex technical task?
- How do you communicate technical risks to non-technical stakeholders?

7. Senior-Level Scenario Prompts

- Design an upgrade-safe extension for a standard S/4HANA process.
- Design a high-volume transactional application using RAP and explain your major decisions.
- A legacy custom solution must move toward a Clean Core approach. What would you assess first?
- Your team has recurring defects in production. How would you improve the development process?

HOW TO PREPARE FOR THE INTERVIEW

Step 1 — Build your foundation

For fresher roles, be able to explain every basic ABAP concept in simple language and write small programs without depending on memorized answers.

Step 2 — Move from definitions to scenarios

With 1–2 years of experience, prepare examples from reports, enhancements, debugging, integrations and production support.

Step 3 — Explain your own project deeply

At 3–5 years, interviewers can ask why you selected a particular technology, how you solved performance issues and what alternatives you considered.

Step 4 — Think like an owner

At 6+ years, prepare for design, architecture, technical trade-offs, governance, mentoring and incident ownership.

Golden rule

Never claim experience you do not have. If you know the concept but have not implemented it, clearly separate theoretical knowledge from hands-on experience.

QUICK REEL VERSION

0–1 YOE: ABAP basics → DDIC → Internal Tables → Open SQL → OOP → Debugging

1–2 YOE: Performance → Enhancements → ALV → BAPI/IDoc/OData → Jobs → CDS/RAP basics

3–5 YOE: Advanced ABAP → CDS → AMDP → RAP → OData/Fiori → Clean Core → Scenarios

6+ YOE: Architecture → RAP design → S/4HANA extensibility → APIs → Performance → Leadership

Save this roadmap before your next SAP ABAP interview.

Use it as a checklist and add questions from your own project experience.