Faysal Aziz

From Oracle & SQL Server to PostgreSQL

Migration, Modernization and Performance

FA-0215 · three-day

DevOps, Cloud & Infrastructure · Data & Analytics

Introduction

Move the data, translate the workload, tune the platform.

Migrating from Oracle 19c or Microsoft SQL Server to PostgreSQL is not simply a matter of copying tables from one database engine to another. The difficult work usually appears around the data: datatype differences, procedural code, sequences and identities, indexing strategies, transaction behavior, optimizer decisions, stored procedures, application SQL, and workloads that performed well on the source platform but behave very differently after conversion.

This three-day course concentrates on those practical migration issues rather than attempting to teach PostgreSQL administration from the ground up. Participants will learn how to assess an existing Oracle 19c or SQL Server workload, identify migration complexity before committing to a strategy, convert schemas and data, deal with incompatible SQL and procedural code, validate the migrated environment, and optimize PostgreSQL for the resulting workload. Particular attention is given to avoiding the common mistake of reproducing Oracle or SQL Server designs literally inside PostgreSQL instead of adapting them to PostgreSQL's architecture and optimizer.

The course reflects the current PostgreSQL ecosystem, including PostgreSQL 18, whose current documentation includes expanded performance, statistics, EXPLAIN, and logical-replication capabilities. For Oracle migrations, the course incorporates current Ora2Pg capabilities for migration assessment, schema and data extraction, PL/SQL conversion, testing, and validation; the current Ora2Pg project lists version 25.0 and includes migration-assessment functionality. For SQL Server workloads, participants will also examine practical migration tooling such as pgloader, which supports direct SQL Server-to-PostgreSQL migration and datatype transformation.

The instructor brings **over 30 years of industry experience** and will approach the course from the perspective of real production migrations. The emphasis will therefore be on industry-demanded skills, migration decisions, troubleshooting, performance and operational readiness rather than academic

database theory. Given the three-day limitation, the course deliberately prioritizes the areas most likely to determine whether a migration succeeds or fails.

Learning outcomes

By the end of this course, participants should be able to:

- Assess Oracle 19c and SQL Server databases for PostgreSQL migration complexity.
- Identify important architectural and SQL-dialect differences between the source platforms and PostgreSQL.
- Select appropriate migration strategies and tools.
- Plan schema, data, code and application migration.
- Map Oracle and SQL Server datatypes appropriately to PostgreSQL.
- Recognize PL/SQL and T-SQL constructs requiring redesign or manual conversion.
- Perform schema and bulk-data migration using appropriate tooling.
- Identify indexing and query-design changes required after migration.
- Analyze PostgreSQL execution plans using EXPLAIN and EXPLAIN ANALYZE.
- Use PostgreSQL statistics and pg_stat_statements to identify expensive SQL workloads.
- Optimize migrated workloads instead of simply reproducing source-platform configurations.
- Validate migrated data, database objects and application behavior.
- Develop practical cutover, rollback and post-migration stabilization strategies.

Prerequisites

Participants should have:

- Deep knowledge of relational databases and SQL.
- Working administration experience with Oracle, SQL Server or PostgreSQL.
- Experience with tables, indexes, views, constraints and stored procedures.
- Good understanding of database performance concepts.
- Advanced Linux or command-line experience.
- No previous PostgreSQL migration experience is required.

Training outline

1. PostgreSQL Migration Architecture and Assessment

1. Oracle 19c, SQL Server and PostgreSQL architectural differences

2. PostgreSQL processes, memory and storage fundamentals

Migration approaches

1. Offline migration
2. Online migration
3. Phased migration
4. Minimal-downtime considerations

Migration discovery and inventory

1. Database objects
2. Data volumes
3. Dependencies
4. Application SQL
5. Procedural code

Migration complexity assessment

1. Ora2Pg assessment
2. SQL Server assessment considerations
3. Migration effort classification
4. Unsupported and high-risk objects

Migration planning

1. Schema migration
2. Data migration
3. Code migration
4. Application migration
5. Cutover and rollback planning

2. Schema and SQL Compatibility Migration

Datatype mapping

1. Oracle-to-PostgreSQL datatype considerations
2. SQL Server-to-PostgreSQL datatype considerations
3. Numeric, character and temporal types
4. LOB and binary data
5. Boolean and UUID considerations

Object conversion

1. Tables and constraints
2. Sequences and identity columns
3. Views and materialized views

4. Indexes
5. Synonyms and database links

SQL dialect conversion

1. Oracle SQL differences
2. T-SQL differences
3. PostgreSQL SQL conventions

Procedural code conversion

1. PL/SQL to PL/pgSQL
2. T-SQL to PL/pgSQL
3. Functions and procedures
4. Triggers
5. Exception handling
6. Transaction differences

3. Migration Tooling and Data Movement

Ora2Pg

1. Configuration and connectivity
2. Assessment reports
3. Schema extraction
4. Data export
5. PL/SQL conversion
6. Migration testing

SQL Server migration tooling

1. pgloader
2. Schema and datatype transformation
3. Bulk data loading

PostgreSQL native utilities

1. COPY
2. pg_dump and pg_restore
3. psql

Large database migration considerations

1. Parallel loading
2. Index and constraint creation timing
3. Transaction sizing
4. LOB handling

5. Error and reject management

Incremental and low-downtime migration concepts

1. Change synchronization
2. CDC considerations
3. Logical replication concepts
4. Final synchronization

4. PostgreSQL Performance Optimization After Migration

PostgreSQL query optimizer fundamentals

1. Cost-based optimization
2. Statistics
3. Cardinality estimation
4. Scan and join strategies

Execution-plan analysis

1. EXPLAIN
2. EXPLAIN ANALYZE
3. Buffers and I/O analysis
4. Estimated versus actual rows

Index migration and redesign

1. B-tree indexing
2. Composite indexes
3. Partial indexes
4. Expression indexes
5. Redundant index identification

Query optimization

1. Migrated SQL anti-patterns
2. Join optimization
3. Subqueries and CTE considerations
4. Sorting and aggregation

Workload analysis

1. pg_stat_statements
2. Expensive-query identification
3. Execution frequency and total workload cost

Statistics and maintenance

1. ANALYZE

2. VACUUM
3. Autovacuum fundamentals
4. Statistics targets

PostgreSQL configuration considerations

1. Memory parameters
2. Planner parameters
3. Connection management
4. WAL and checkpoint considerations

5. Migration Validation and Production Cutover

Schema validation

1. Object comparison
2. Constraints and indexes
3. Procedural objects

Data validation

1. Row counts
2. Data consistency
3. Sequence and identity verification
4. LOB validation

Functional validation

1. Application connectivity
2. SQL behavior
3. Transaction behavior
4. Permissions and roles

Performance validation

1. Baseline comparison
2. Critical SQL verification
3. Load and concurrency considerations
4. Post-migration regression identification

Production cutover

1. Migration freeze
2. Final synchronization
3. Application switching
4. Post-cutover verification
5. Rollback readiness

Post-migration stabilization

1. Query monitoring
2. Statistics refresh
3. Index review
4. Autovacuum monitoring
5. Performance remediation
6. Migration sign-off

Disclaimer

This training outline is intended as a structured guideline for a three-day professional course. Actual coverage, sequence, depth and emphasis may be adjusted by the trainer based on participant experience, source-system complexity, available laboratory environments, organizational requirements and developments in the relevant technologies. The trainer therefore reserves the right to amend, substitute, combine or omit topics where professionally appropriate without prior notice, while maintaining the overall objectives of the course.

© 2026 Faysal Aziz · Kuala Lumpur