

- نام دوره: Oracle PL/SQL
- مدت دوره: 40 ساعت
- پیش نیاز: Oracle SQL
- منابع درسی: اسلاید و منابع شرکت اوراکل
- خلاصه

دانشجویان پس از طی این دوره شما قادر خواهند بود توسط زبان PL/SQL در پایگاه داده اوراکل زیر برنامه هایی از قبیل procedure, trigger, function, package و غیره را ایجاد کنند و با امکانات اوراکل که مربوط به بهبود کارایی پایگاه داده و بهبود عملکرد پرس و جوها می باشد از جمله indexing, partitioning, and materialized views summary data و performance tuning آشنا خواهند شد

### ➤ مخاطبین

- ✓ برنامه نویسان اوراکل که از پایگاه داده اوراکل جهت پیاده سازی کسب و کار برنامه های خود استفاده می کنند
- ✓ برنامه نویسان زبانهای گوناگون (java, .net و غیره) که جهت نگهداری اطلاعات برنامه های خود از پایگاه داده اوراکل استفاده میکنند
- ✓ طراحان و برنامه نویسان انبار داده (Data Warehouse) که برای پیاده سازی ETL های خود از پایگاه داده اوراکل استفاده میکنند
- ✓ کارفرمایانی که از پایگاه داده اوراکل استفاده کرده و نیاز به کسب دانش بیشتر جهت استخراج و کار با داده های جمع آوری شده توسط برنامه های کاربردی دارند

### ➤ اهداف دوره

- ✓ رویه و تابع ایجاد کرده و روشهای مختلف استفاده از آنها را بشناسند
- ✓ انواع پکیج ها را طراحی و ایجاد کنند
- ✓ با انواع پکیج های از پیش ساخته اوراکل آشنا شده و از آنها در نوشتن برنامه های خود استفاده کنند
- ✓ از روشهای overload رویه ها و توابع برای انعطاف پذیری بیشتر برنامه های کاربردی استفاده کنند
- ✓ برای پوشش کسب و کارهای برنامه های کاربردی trigger تعریف کنند
- ✓ با روش های گوناگون پیاده سازی SQL های پویا آشنا شوند
- ✓ وابستگی های موجود در پایگاه داده را شناخته و مدیریت کنند
- ✓ از انتخاب های مربوط به بهبود کارایی از جمله پارتیشن بندی، ایندکس گذاری، خلاصه سازی داده ها استفاده کنید
- ✓ از امکانات پیشرفته دستورات SQL اوراکل از جمله توابع تحلیلی، پرس و جوهای بازگشتی، Temporary Table استفاده کنید

### ➤ نام فارسی سرفصل ها

- ✓ معرفی PL/SQL
- ✓ تعریف کردن متغیرهای PL/SQL
- ✓ نوشتن دستورات قابل اجرا

- ✓ تعامل با پایگاه داده اوراکل: دستورات SQL در برنامه های PL/SQL
- ✓ نوشتن ساختارهای کنترلی
- ✓ کارکردن با نوع های داده ای ترکیبی
- ✓ استفاده کردن از cursor های صریح
- ✓ مدیریت کردن خطاها
- ✓ ایجاد کردن رویه ها
- ✓ ایجاد کردن خطاها و خطایابی زیربرنامه ها
- ✓ ایجاد کردن پکیج ها
- ✓ استفاده کردن از پکیج های ارائه شده توسط اوراکل در پیاده سازی برنامه های کاربردی
- ✓ استفاده کردن از SQL پویا
- ✓ در نظر گرفتن ملاحظات برای کدهای PL/SQL
- ✓ ایجاد کردن تریگرها
- ✓ ایجاد کردن تریگرهای برای عملیات ترکیبی ، DDL ها و اتفاقات پایگاه داده
- ✓ بهبود کارایی

## **PL/SQL Fundamental Objectives**

- Introduction to PL/SQL
- Declaring PL/SQL Identifiers
- Writing Executable Statements
- Interacting with the Oracle Server
- Writing Control Structures
- Working with Composite Data Types
- Using Cursors
- Handling Exceptions
- Creating Stored Procedures and Functions

## **Develop PL/SQL with Program Units and performance tuning Objectives**

### **Creating Stored Procedures**

- Describe PL/SQL blocks and subprograms
- Describe the uses and benefits of procedures
- Create, call, and remove procedures
- Use formal and actual parameters
- Identify the available parameter-passing modes
- Pass parameters using the positional, named, or combination techniques
- Handle exceptions in procedures
- View the procedure information

## **Creating Functions and Debugging Subprograms**

Creating Stored Functions

The Difference Between Procedures and Functions

Developing Functions

Creating and Executing and Removing Functions

Identifying the Advantages of Using Stored Functions in SQL Statements

Using User-Defined Functions in SQL Statements

Using a PL/SQL Function in the SQL WITH Clause

□ Restrictions When Calling Functions from SQL statements

## **Creating Packages**

Using PL/SQL Packages

The Components of a PL/SQL Package

The Visibility of a Package's Components

Developing a PL/SQL Package

Creating the Package Specification and Package Body

Invoking the Package Constructs

Creating and Using Bodiless Packages

Removing a Package

## **Working with Packages**

Overloading Subprograms

Using Forward Declarations to Solve Illegal Procedure Reference

Initializing Packages

Using Package Functions in SQL and Restrictions

Controlling Side Effects of PL/SQL Subprograms

Persistent State of Packages

Persistent State of Package Variables and Cursors

Using PL/SQL Tables of Records in Packages

## **Using Oracle-Supplied Packages in Application Development**

Using Oracle-Supplied Packages

Examples of Some of the Oracle-Supplied Packages

How Does the DBMS\_OUTPUT Package Work?

Using the UTL\_FILE Package to Interact With Operating System Files

Using the UTL\_MAIL Package

## **Using Dynamic SQL**

The Execution Flow of SQL

Working With Dynamic SQL

When Do You Need Dynamic SQL?

Using Native Dynamic SQL (NDS)

Declaring Cursor Variables

Executing a PL/SQL Block Dynamically

□ Using Native Dynamic SQL to Compile PL/SQL Code

## **Design Considerations for PL/SQL Code**

Standardize constants with a constant package

- Standardize exceptions with an exception package
- Write PL/SQL code that uses local subprograms
- Grant Roles to PL/SQL Packages and Standalone Stored Subprograms
- Use the NOCOPY compiler hint to pass parameters by reference
- Use the PARALLEL ENABLE hint for optimization
- Use the AUTONOMOUS TRANSACTION pragma to run independent transactions within a single transaction
- Describe the differences between invoker rights and definer rights

## **Creating Compound, DDL, and Event Database Triggers**

- Describe compound triggers
- Describe mutating tables
- Create triggers on DDL statements
- Create triggers on system events
- Display information about triggers

## **Oracle Index Optimization**

- Oracle Index types
- Index cardinality
- multi-column indexes
- bitmap indexes
- Index unique vs. index normal
- Finding missing indexes
- Partition Index and Local Indexes

## **Partitioning**

- Getting optimal pruning
- Partition exchange loading
- Partitioning and unusable indexes
- Partitioning method
- Partitioning key
- Partition granularity
- Partitioning Indexes
- Local Prefixed Indexes
- Local Non-Prefixed Indexes
- Global Prefixed Indexes
- Global Non-Prefixed Indexes
- Efficient statistics management

## **Materialized view**

- Explain the benefits of materialized views
- Increase application performance using Materialized Views
- Learn how to create and use Materialized
- Different types of materialized views
- Views and Materialized View logs
- Set Materialized Views to refresh in a variety of ways
- Create materialized views
- Create materialized view logs

- Refreshing materialized views
- Altering Materialized Views
- Dropping Materialized views
- Understand Materialized View Groups
- Exploit Oracle's Query Rewrite capability

### **Advanced Features For Query and Query Performance Tuning Tips**

- Oracle SQL Analytics
- SQL Aggregate Functions
- Oracle Hierarchies
- Using With Clause

FardOracle