

# SQL SERVER COURSE SYLLABUS

## Review of Basic SQL and history of SQL:

• Introduction

• History of SQL

• Evolution of SQL

• Select

• Project

• Join

• Describing Oracle tables

• Restricting row returns

• Where clause

• Complex Boolean logic in SQL

• ROWID restrictions

## Entity/relation modeling

Types of data relationships

Data normalization

[One-to-one relationships](#)

[One-to-many relationships](#)

[Many-to-many relationships](#)

[Recursive many-to-many relationships](#)

## Optimizing Oracle SQL

[Steps in SQL optimization Parsing a SQL statement](#)

[First rows vs all rows optimization Oracle optimizer overview](#)

[Management issues with system-wide optimization Different modes of SQL optimization](#)

[Bi-modal databases Rule-based optimization Cost-based optimization All rows optimization First rows optimization](#)

- [SQL Semantic Analysis](#)

- [Generating the execution plan](#)

- [Using optimizer plan stability](#)

- [Using the v\\$sql view](#)

- [Using the v\\$sql\\_plan view](#)

- [Using SQL\\*Plus](#)

- [Creating basic reports](#)

- [Using the set commands](#)

- [Adjusting line output](#)

- [Setting pagesize and linesize](#)

- [Echo](#)

- [Feedback](#)

- [Feedback](#)

- [termout](#)

- [Column wrapping](#)

- [Creating breaks and summaries](#)

- [Adding prompts to queries](#)

## Explain plan and reading execution plans

- [Explain plan](#)

- [Oracle autotrace](#)

- [How to read an execution plan](#)

## Altering SQL execution plans

- [Using hints](#)

## Table joining internals

- [Sort-merge joins](#)

- [Nested Loop joins](#)

- [Hash joins](#)

- [STAR joins](#)

- [Bitmap joins](#)

- [Equi-join](#)

- [Outer join](#)

- [Hiding joins by creating views](#)

## Subqueries

- [Using IN, NOT IN, EXISTS and NOT EXISTS](#)

- [Subqueries](#)

- [Correlated subquery](#)

- [Non-correlated subqueries](#)

## Advanced SQL operators

- [Between operator](#)

- [IN and NOT In operators](#)

- [Sub-queries](#)

- [EXISTS clause](#)

- [Using wildcards in queries \(LIKE operator\)](#)

## SQL access methods

- [Review of Basic joining methods](#)

- [Merge join](#)

- [Hash Join](#)

- [Nested Loop join](#)

Advanced SQL operators

Between operator

## SQL Tuning

### DML and SQL Tuning

Writing and optimizing INSERT statements

Writing and optimizing DELETE statements

• Writing and optimizing DELETE statements

## Optimizer Statistics

Purpose of statistics

Types of statistics (table, column, system) Histogram statistics

Dynamic sampling using dbms\_stats

Exporting/importing statistics

Statistics management

## SQL Tuning and full-table scans

Basics of file I/O

Sequential reads vs. scattered reads

When full scans are best

RAM caching in the SGA Automating table caching

Solid State Disks

## Aggregation in SQL

Count(\*)

Sum

Avg

Min and max

Using the group by clause

## PL/SQL Section

### Basics of PL/SQL

PL/SQL architecture

PL/SQL and SQL\*Plus

PL/SQL Basics

Introduction to PL/SQL

PL/SQL as a 4th generation language

Compiling vs. Interpreting

Declare statement

Using PL/SQL Variables

PL/SQL Constants

PL/SQL Datatypes

## PL/SQL structures

Simple blocks

Control structures

PL/SQL records

Recognizing the Basic PL/SQL Block and Its Sections

Describing the Significance of Variables in PL/SQL

Distinguishing Between PL/SQL and Non-PL/SQL Variables

Declaring Variables and Constants

Executing a PL/SQL Block

## Error checking – exception handling

Defining exceptions

Using the when others clause

Ensuring complete error checking

Passing error messages to calling routine

## Boolean logic in PL/SQL

• Identifying the Uses and Types of Control Structures

• Constructing an IF Statement

• Constructing and Identifying Different Loop Statements

• Controlling Block Flow Using Nested Loops and Labels

• Using Logic Tables

- If-then-else structure

- Testing for numbers characters and Booleans

## Cursors in PL/SQL

Cursor basics

Using a cursor for a multi-row SQL query

## Iteration in PL/SQL

For loop

While loop

## PL/SQL tables

Defining PL/SQL tables

Reasons to use PL/SQL tables

Populating a PL/SQL table

Retrieving from a PL/SQL table

## Dynamic SQL in PL/SQL

Introduction to the dbms\_sql package

Creating a dynamic SQL statement

## Nested blocks in PL/SQL

Creating nested blocks



[Understanding scope in nested blocks](#)

## Triggers in PL/SQL

[Triggers and database events](#)

[Defining a trigger](#)

[Timing a trigger](#)

[Enabling and disabling a trigger](#)

## Stored procedures, functions and packages

• [Basics of stored procedures](#)

• [Basics of functions](#)

• [Basics of packages](#)

• [Defining stored procedures & functions](#)

• [Function and stored procedures prototypes](#)

• [Passing arguments to functions and stored procedures](#)

• [Package forward declaration](#)

• [Package dependency](#)

• [Package overloading](#)

• [Listing package information](#)

• [Recompiling functions and stored procedures](#)

• [Pinning packages in the SGA with dbms\\_shared\\_pool.keep](#)

## Bulking in PL/SQL

Bulk queries

Bulk DML (forall statement)

Using cursor attributes