**Introduction to Oracle9*i*: SQL**

**Student Guide • Volume 1**

40049GC10

Production 1.0

June 2001

D33051

**Authors**

Nancy Greenberg

Priya Nathan

**Technical Contributors and Reviewers**

Josephine Turner

Anna Atkinson

Don Bates

Marco Berbeek

Andrew Brannigan

Michael Gerlach

Sharon Gray

Rosita Hanoman

Mozhe Jalali

Sarah Jones

Charbel Khouri

Christopher Lawless

Diana Lorentz

Nina Minchen

Cuong Nguyen

Daphne Nougier

Patrick Odell

Laura Pezzini

Stacey Procter

Maribel Renau

Bryan Roberts

Sunshine Salmon

Casa Sharif

Bernard Soleillant

Ruediger Steffan

Karla Villasenor

Andree Wheeley

Lachlan Williams

**Publisher**

Sheryl Domingue

**Copyright © Oracle Corporation, 2000, 2001. All rights reserved.**

This documentation contains proprietary information of Oracle Corporation. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

**Restricted Rights Legend**

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with “Restricted Rights,” as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Education Products, Oracle Corporation, 500 Oracle Parkway, Box SB-6, Redwood Shores, CA 94065. Oracle Corporation does not warrant that this document is error-free.

Oracle and all references to Oracle products are trademarks or registered trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

**Contents**

**Preface**

**Curriculum Map**

**Introduction**

Objectives I-2

Oracle9*i* I-3

Oracle9*i* Application Server I-5

Oracle9*i* Database I-6

Oracle9*i*: Object Relational Database Management System I-8 Oracle Internet Platform I-9

System Development Life Cycle I-10

Data Storage on Different Media I-12

Relational Database Concept I-13

Definition of a Relational Database I-14

Data Models I-15

Entity Relationship Model I-16

Entity Relationship Modeling Conventions I-17

Relating Multiple Tables I-19

Relational Database Terminology I-20

Relational Database Properties I-21

Communicating with a RDBMS Using SQL I-22

Relational Database Management System I-23

SQL Statements I-24

Tables Used in the Course I-25

Summary I-26

**1 Writing Basic SQL SELECT Statements**

Objectives 1-2

Capabilities of SQL SELECT Statements 1-3

Basic SELECT Statement 1-4

**iii**

Selecting All Columns 1-5

Selecting Specific Columns 1-6

Writing SQL Statements 1-7

Column Heading Defaults 1-8

Arithmetic Expressions 1-9

Using Arithmetic Operators 1-10

Operator Precedence 1-11

Using Parentheses 1-13

Defining a Null Value 1-14

Null Values in Arithmetic Expressions 1-15 Defining a Column Alias 1-16

Using Column Aliases 1-17

Concatenation Operator 1-18

Using the Concatenation Operator 1-19

Literal Character Strings 1-20

Using Literal Character Strings 1-21

Duplicate Rows 1-22

Eliminating Duplicate Rows 1-23

SQL and iSQL\*Plus Interaction 1-24

SQL Statements versus iSQL\*Plus Commands 1-25 Overview of iSQL\*Plus 1-26

Logging In to iSQL\*Plus 1-27

The iSQL\*Plus Environment 1-28

Displaying Table Structure 1-29

Interacting with Script Files 1-31

Summary 1-34

Practice 1 Overview 1-35

**iv**

**2 Restricting and Sorting Data** Objectives 2-2

Limiting Rows Using a Selection 2-3 Limiting the Rows Selected 2-4

Using the WHERE Clause 2-5

Character Strings and Dates 2-6

Comparison Conditions 2-7

Using Comparison Conditions 2-8 Other Comparison Conditions 2-9 Using the BETWEEN Condition 2-10 Using the IN Condition 2-11

Using the LIKE Condition 2-12

Using the NULL Conditions 2-14

Logical Conditions 2-15

Using the AND Operator 2-16

Using the OR Operator 2-17

Using the NOT Operator 2-18

Rules of Precedence 2-19

ORDER BY Clause 2-22

Sorting in Descending Order 2-23 Sorting by Column Alias 2-24

Sorting by Multiple Columns 2-25 Summary 2-26

Practice 2 Overview 2-27

**v**

**3 Single-Row Functions**

Objectives 3-2

SQL Functions 3-3

Two Types of SQL Functions 3-4

Single-Row Functions 3-5

Character Functions 3-7

Case Manipulation Functions 3-9

Using Case Manipulation Functions 3-10 Character-Manipulation Functions 3-11

Using the Character-Manipulation Functions 3-12 Number Functions 3-13

Using the ROUND Function 3-14

Using the TRUNC Function 3-15

Using the MOD Function 3-16

Working with Dates 3-17

Arithmetic with Dates 3-19

Using Arithmetic Operators with Dates 3-20 Date Functions 3-21

Using Date Functions 3-22

Practice 3, Part 1 Overview 3-24

Conversion Functions 3-25

Implicit Data-Type Conversion 3-26

Explicit Data-Type Conversion 3-28

Using the TO\_CHAR Function with Dates 3-31 Elements of the Date Format Model 3-32 Using the TO\_CHAR Function with Dates 3-36

**vi**

Using the TO\_CHAR Function with Numbers 3-37 Using the TO\_NUMBER and TO\_DATE Functions 3-39 RR Date Format 3-40

Example of RR Date Format 3-41

Nesting Functions 3-42

General Functions 3-44

NVL Function 3-45

Using the NVL Function 3-46

Using the NVL2 Function 3-47

Using the NULLIF Function 3-48

Using the COALESCE Function 3-49

Conditional Expressions 3-51

The CASE Expression 3-52

Using the CASE Expression 3-53

The DECODE Function 3-54

Using the DECODE Function 3-55

Summary 3-57

Practice 3, Part 2 Overview 3-58

**4 Displaying Data from Multiple Tables**

Objectives 4-2

Obtaining Data from Multiple Tables 4-3

Cartesian Products 4-4

Generating a Cartesian Product 4-5

Types of Joins 4-6

Joining Tables Using Oracle Syntax 4-7

**vii**

What Is an Equijoin? 4-8

Retrieving Records with Equijoins 4-9

Additional Search Conditions Using the AND Operator 4-10 Qualifying Ambiguous Column Names 4-11

Using Table Aliases 4-12

Joining More than Two Tables 4-13

Nonequijoins 4-14

Retrieving Records with Nonequijoins 4-15

Outer Joins 4-16

Outer Joins Syntax 4-17

Using Outer Joins 4-18

Self Joins 4-19

Joining a Table to Itself 4-20

Practice 4, Part 1 Overview 4-21

Joining Tables Using SQL: 1999 Syntax 4-22 Creating Cross Joins 4-23

Creating Natural Joins 4-24

Retrieving Records with Natural Joins 4-25

Creating Joins with the USING Clause 4-26

Retrieving Records with the USING Clause 4-27 Creating Joins with the ON Clause 4-28

Retrieving Records with the ON Clause 4-29

Creating Three-Way Joins with the ON Clause 4-30 INNER versus OUTER Joins 4-31

LEFT OUTER JOIN 4-32

RIGHT OUTER JOIN 4-33

**viii**

FULL OUTER JOIN 4-34

Additional Conditions 4-35

Summary 4-36

Practice 4, Part 2 Overview 4-37

**5 Aggregating Data Using Group Functions** Objectives 5-2

What Are Group Functions? 5-3

Types of Group Functions 5-4

Group Functions Syntax 5-5

Using the AVG and SUM Functions 5-6

Using the MIN and MAX Functions 5-7

Using the COUNT Function 5-8

Using the DISTINCT Keyword 5-10

Group Functions and Null Values 5-11

Using the NVL Function with Group Functions 5-12 Creating Groups of Data 5-13

Creating Groups of Data: GROUP BY Clause Syntax 5-14 Using the GROUP BY Clause 5-15

Grouping by More Than One Column 5-17

Using the GROUP BY Clause on Multiple Columns 5-18 Illegal Queries Using Group Functions 5-19

Excluding Group Results 5-21

Excluding Group Results: The HAVING Clause 5-22 Using the HAVING Clause 5-23

Nesting Group Functions 5-25

Summary 5-26

Practice 5 Overview 5-27

**ix**

**6 Subqueries**

Objectives 6-2

Using a Subquery to Solve a Problem 6-3

Subquery Syntax 6-4

Using a Subquery 6-5

Guidelines for Using Subqueries 6-6

Types of Subqueries 6-7

Single-Row Subqueries 6-8

Executing Single-Row Subqueries 6-9

Using Group Functions in a Subquery 6-10

The HAVING Clause with Subqueries 6-11

What Is Wrong with This Statement? 6-12

Will This Statement Return Rows? 6-13

Multiple-Row Subqueries 6-14

Using the ANY Operator in Multiple-Row Subqueries 6-15 Using the ALL Operator in Multiple-Row Subqueries 6-16 Null Values in a Subquery 6-17

Summary 6-18

Practice 6 Overview 6-19

**7 Producing Readable Output with iSQL\*Plus** Objectives 7-2

Substitution Variables 7-3

Using the & Substitution Variable 7-5

Character and Date Values with Substitution Variables 7-7 Specifying Column Names, Expressions, and Text 7-8

**x**

Defining Substitution Variables 7-10

DEFINE and UNDEFINE Commands 7-11

Using the DEFINE Command with & Substitution Variable 7-12 Using the VERIFY Command 7-14

Customizing the iSQL\*Plus Environment 7-15

SET Command Variables 7-16

iSQL\*Plus Format Commands 7-17

The COLUMN Command 7-18

Using the COLUMN Command 7-19

COLUMN Format Models 7-20

Using the BREAK Command 7-21

Using the TTITLE and BTITLE Commands 7-22

Creating a Script File to Run a Report 7-23

Sample Report 7-25

Summary 7-26

Practice 7 Overview 7-27

**8 Manipulating Data**

Objectives 8-2

Data Manipulation Language 8-3

Adding a New Row to a Table 8-4

The INSERT Statement Syntax 8-5

Inserting New Rows 8-6

Inserting Rows with Null Values 8-7

Inserting Special Values 8-8

Inserting Specific Date Values 8-9

**xi**

Creating a Script 8-10

Copying Rows from Another Table 8-11

Changing Data in a Table 8-12

The UPDATE Statement Syntax 8-13

Updating Rows in a Table 8-14

Updating Two Columns with a Subquery 8-15

Updating Rows Based on Another Table 8-16

Updating Rows: Integrity Constraint Error 8-17

Removing a Row from a Table 8-18

The DELETE Statement 8-19

Deleting Rows from a Table 8-20

Deleting Rows Based on Another Table 8-21

Deleting Rows: Integrity Constraint Error 8-22

Using a Subquery in an INSERT Statement 8-23

Using the WITH CHECK OPTION Keyword on DML Statements 8-25 Overview of the Explict Default Feature 8-26

Using Explicit Default Values 8-27

The MERGE Statement 8-28

MERGE Statement Syntax 8-29

Merging Rows 8-30

Database Transactions 8-32

Advantages of COMMIT and ROLLBACK Statements 8-34 Controlling Transactions 8-35

Rolling Back Changes to a Marker 8-36

Implicit Transaction Processing 8-37

State of the Data Before COMMIT or ROLLBACK 8-38

State of the Data After COMMIT 8-39

**xii**

Committing Data 8-40

State of the Data After ROLLBACK 8-41

Statement-Level Rollback 8-42

Read Consistency 8-43

Implementation of Read Consistency 8-44 Locking 8-45

Implicit Locking 8-46

Summary 8-47

Practice 8 Overview 8-48

**9 Creating and Managing Tables**

Objectives 9-2

Database Objects 9-3

Naming Rules 9-4

The CREATE TABLE Statement 9-5

Referencing Another User’s Tables 9-6

The DEFAULT Option 9-7

Creating Tables 9-8

Tables in the Oracle Database 9-9

Querying the Data Dictionary 9-10

Data Types 9-11

Datetime Data Types 9-13

TIMESTAMP WITH TIME ZONE Data Type 9-15 TIMESTAMP WITH LOCAL TIME Data Type 9-16 INTERVAL YEAR TO MONTH Data Type 9-17 Creating a Table by Using a Subquery Syntax 9-18

**xiii**

Creating a Table by Using a Subquery 9-19 The ALTER TABLE Statement 9-20

Adding a Column 9-22

Modifying a Column 9-24

Dropping a Column 9-25

The SET UNUSED Option 9-26

Dropping a Table 9-27

Changing the Name of an Object 9-28

Truncating a Table 9-29

Adding Comments to a Table 9-30

Summary 9-31

Practice 9 Overview 9-32

**10 Including Constraints**

Objectives 10-2

What Are Constraints? 10-3

Constraint Guidelines 10-4

Defining Constraints 10-5

The NOT NULL Constraint 10-7

The UNIQUE Constraint 10-9

The PRIMARY KEY Constraint 10-11

The FOREIGN KEY Constraint 10-13

FOREIGN KEY Constraint Keywords 10-15 The CHECK Constraint 10-16

Adding a Constraint Syntax 10-17

Adding a Constraint 10-18

Dropping a Constraint 10-19

**xiv**

Disabling Constraints 10-20

Enabling Constraints 10-21

Cascading Constraints 10-22

Viewing Constraints 10-24

Viewing the Columns Associated with Constraints 10-25 Summary 10-26

Practice 10 Overview 10-27

**11 Creating Views**

Objectives 11-2

Database Objects 11-3

What Is a View? 11-4

Why Use Views? 11-5

Simple Views and Complex Views 11-6

Creating a View 11-7

Retrieving Data from a View 11-10

Querying a View 11-11

Modifying a View 11-12

Creating a Complex View 11-13

Rules for Performing DML Operations on a View 11-14 Using the WITH CHECK OPTION Clause 11-17 Denying DML Operations 11-18

Removing a View 11-20

Inline Views 11-21

Top-n Analysis 11-22

Performing Top-n Analysis 11-23

**xv**

Example of Top-n Analysis 11-24

Summary 11-25

Practice 11 Overview 11-26

**12 Other Database Objects**

Objectives 12-2

Database Objects 12-3

What Is a Sequence? 12-4

The CREATE SEQUENCE Statement Syntax 12-5 Creating a Sequence 12-6

Confirming Sequences 12-7

NEXTVAL and CURRVAL Pseudocolumns 12-8 Using a Sequence 12-10

Modifying a Sequence 12-12

Guidelines for Modifying a Sequence 12-13 Removing a Sequence 12-14

What Is an Index? 12-15

How Are Indexes Created? 12-16

Creating an Index 12-17

When to Create an Index 12-18

When Not to Create an Index 12-19

Confirming Indexes 12-20

Function-Based Indexes 12-21

Removing an Index 12-22

Synonyms 12-23

**xvi**

Creating and Removing Synonyms 12-24

Summary 12-25

Practice 12 Overview 12-26

**13 Controlling User Access**

Objectives 13-2

Controlling User Access 13-3

Privileges 13-4

System Privileges 13-5

Creating Users 13-6

User System Privileges 13-7

Granting System Privileges 13-8

What Is a Role? 13-9

Creating and Granting Privileges to a Role 13-10

Changing Your Password 13-11

Object Privileges 13-12

Granting Object Privileges 13-14

Using the WITH GRANT OPTION and PUBLIC Keywords 13-15 Confirming Privileges Granted 13-16

How to Revoke Object Privileges 13-17

Revoking Object Privileges 13-18

Database Links 13-19

Summary 13-21

Practice 13 Overview 13-22

**14 SQL Workshop Workshop Overview**

Workshop Overview 14-2

**xvii**

**15 Using SET Operators**

Objectives 15-2

The SET Operators 15-3

Tables Used in This Lesson 15-4

The UNION SET Operator 15-7

Using the UNION Operator 15-8

The UNION ALL Operator 15-10

Using the UNION ALL Operator 15-11

The INTERSECT Operator 15-12

Using the INTERSECT Operator 15-13 The MINUS Operator 15-14

SET Operator Guidelines 15-16

The Oracle Server and SET Operators 15-17 Matching the SELECT Statements 15-18 Controlling the Order of Rows 15-20

Summary 15-21

Practice 15 Overview 15-22

**16 Oracle 9*i* Datetime Functions**

Objectives 16-2

TIME ZONES 16-3

Oracle 9*i* Datetime Support 16-4

CURRENT\_DATE 16-6

CURRENT\_TIMESTAMP 16-7

LOCALTIMESTAMP 16-8

DBTIMEZONE and SESSIONTIMEZONE 16-9 **xviii**

EXTRACT 16-10

FROM\_TZ 16-11

TO\_TIMESTAMP and TO\_TIMESTAMP\_TZ 16-12 TO\_YMINTERVAL 16-13

TZ\_OFFSET 16-14

Summary 16-16

Practice 16 Overview 16-17

**17 Enhancements to the GROUP BY Clause** Objectives 17-2

Review of Group Functions 17-3

Review of the GROUP BY Clause 17-4

Review of the HAVING Clause 17-5

GROUP BY with ROLLUP and CUBE Operators 17-6 ROLLUP Operator 17-7

ROLLUP Operator Example 17-8

CUBE Operator 17-9

CUBE Operator: Example 17-10

GROUPING Function 17-11

GROUPING Function: Example 17-12

GROUPING SETS 17-13

GROUPING SETS: Example 17-15

Composite Columns 17-17

Composite Columns: Example 17-19

Concatenated Groupings 17-21

**xix**

Concatenated Groupings Example 17-22 Summary 17-23

Practice 17 Overview 17-24

**18 Advanced Subqueries**

Objectives 18-2

What Is a Subquery? 18-3

Subqueries 18-4

Using a Subquery 18-5

Multiple-Column Subqueries 18-6

Column Comparisons 18-7

Pairwise Comparison Subquery 18-8

Nonpairwise Comparison Subquery 18-9 Using a Subquery in the FROM Clause 18-10 Scalar Subquery Expressions 18-11

Correlated Subqueries 18-14

Using Correlated Subqueries 18-16

Using the EXISTS Operator 18-18

Using the NOT EXISTS Operator 18-20 Correlated UPDATE 18-21

Correlated DELETE 18-24

The WITH Clause 18-26

WITH Clause: Example 18-27

Summary 18-29

Practice 18 Overview 18-31

**xx**

**19 Hierarchical Retrieval**

Objectives 19-2

Sample Data from the EMPLOYEES Table 19-3

Natural Tree Structure 19-4

Hierarchical Queries 19-5

Walking the Tree 19-6

Walking the Tree: From the Bottom Up 19-8

Walking the Tree: From the Top Down 19-9

Ranking Rows with the LEVEL Pseudocolumn 19-10 Formatting Hierarchical Reports Using LEVEL and LPAD 19-11 Pruning Branches 19-13

Summary 19-14

Practice 19 Overview 19-15

**20 Oracle 9*i* Extensions to DML and DDL Statements** Objectives 20-2

Review of the INSERT Statement 20-3

Review of the UPDATE Statement 20-4

Overview of Multitable INSERT Statements 20-5

Types of Multitable INSERT Statements 20-7

Multitable INSERT Statements 20-8

Unconditional INSERT ALL 20-10

Conditional INSERT ALL 20-11

Conditional FIRST INSERT 20-13

Pivoting INSERT 20-15

External Tables 20-18

**xxi**

Creating an External Table 20-19

Example of Creating an External Table 20-20 Querying External Tables 20-23

CREATE INDEX with CREATE TABLE Statement 20-24 Summary 20-25

Practice 20 Overview 20-26

**A Practice Solutions**

**B Table Descriptions and Data**

**C Using SQL\* Plus**

**D Writing Advanced Scripts**

**E Oracle Architectural Components**

**Index**

**Additional Practices**

**Additional Practice Solutions**

**Table and Descriptions**

**xxii**

**Preface**

**Profile**

**Before You Begin This Course**

Before you begin this course, you should be able to use a graphical user interface (GUI)*.* Required prerequisites are familiarity with data processing concepts and techniques.

**How This Course Is Organized**

*Introduction to Oracle9i: SQL* is an instructor-led course featuring lectures and hands-on exercises. Online demonstrations and written practice sessions reinforce the concepts and skills introduced.

**Preface-3**

**Related Publications**

**Oracle Publications**

**Title Part Number** *Oracle9i Reference, Release 1 (9.0.1)* A90190-01 *Oracle9i SQL Reference, Release 1 (9.0.1)* A90125-01 *Oracle9i Concepts*, *Release 1 (9.0.0)* A88856-01 *Oracle9i Server Application Developer’s*

*Guide Fundamentals, Release 1 (9.0.1)* A88876-01 *iSQL\*Plus User’s Guide and Reference, Release 9.0.0*

*SQL\*Plus User’s Guide and Reference, Release 9.0.1* A88827-01

**Additional Publications**

• System release bulletins

• Installation and user’s guides

• read.me files

• International Oracle User’s Group (IOUG) articles

• *Oracle Magazine*

**Preface-4**

**Typographic Conventions**

What follows are two lists of typographical conventions used specifically within text or within code.

**Typographic Conventions within Text**

**Convention Object or Term Example**

Uppercase Commands, Use the SELECT command to view functions, information stored in the LAST\_NAME

column names, column of the EMPLOYEES table.

table names,

PL/SQL objects,

schemas

Lowercase, Filenames, **where:** *role* is the name of the role italic syntax variables, to be created. usernames,

passwords

Initial cap Trigger and Assign a When-Validate-Item trigger to button names the ORD block.

Choose Cancel.

Italic Books, names of For more information on the subject see the courses and *Oracle Server SQL Language Reference*

manuals, and *Manual*

emphasized

words or phrases Do *not* save changes to the database.

Quotation marks Lesson module This subject is covered in Lesson 3, titles referenced “Working with Objects.”

within a course

**Preface-5**

**Typographic Conventions (continued)**

**Typographic Conventions within Code**

**Convention Object or Term Example**

Uppercase Commands, **SELECT employee\_id**

functions **FROM employees;**

Lowercase, Syntax variables **CREATE ROLE *role*;**

italic

Initial cap Forms triggers **Form module: ORD**

**Trigger level: S\_ITEM.QUANTITY**

**item**

**Trigger name: When-Validate-Item**

**. . .**

Lowercase Column names, **. . .**

table names, **OG\_ACTIVATE\_LAYER**

filenames, **(OG\_GET\_LAYER ('prod\_pie\_layer'))** PL/SQL objects **. . .**

**SELECT last\_name**

**FROM employees;**

Bold Text that must **CREATE USER scott**

be entered by a **IDENTIFIED BY tiger;**

user

**Preface-6**

**Curriculum Map**

**Languages Curriculum for Oracle9*i***

**Introduction to Oracle9*i*: SQL**

**or**

**Introduction to Oracle9*i* for**

**Experienced SQL Users**

**SQL for End Users**

**Extended Data inClass Retrieval**

**SQL1 with SQL inClass**

**inClass**

**Introduction to Oracle9*i*: PL/SQL**

**Develop PL/SQL**

**Program Units PL/SQL**

**Fundamentals**

**inClass**

**Advanced PL/SQL**

**inClass**

Copyright © Oracle Corporation, 2001. All rights reserved.

**Integrated Languages Curriculum**

*Introduction to Oracle9i: SQL* consists of two modules, *SQL1* and *Extended Data Retrieval with SQL*. *SQL1* covers creating database structures and storing, retrieving, and manipulating data in a relational database. *Extended Data Retrieval with SQL* covers advanced SELECT statements, Oracle SQL, and *i*SQL\*Plus Reporting.

For people who have worked with other relational databases and have knowledge of SQL, another course, called *Introduction to Oracle9i for Experienced SQL Users* is offered. This course covers the SQL statements that are not part of ANSI SQL but are specific to Oracle.

*Introduction to Oracle9i: PL/SQL* consists of two modules, *PL/SQL Fundamentals* and *Develop PL/SQL Program Units*. *PL/SQL Fundamentals* covers PL/SQL basics including the PL/SQL language structure, flow of execution and interface with SQL. *Develop PL/SQL Program Units* covers how to create stored procedures, functions, packages, and triggers as well as maintaining and debugging program code.

*SQL for End Users* is geared towards individuals with little programming background and covers the basic SQL statements. This course is for end users that need to know some basic SQL programming.

*Advanced PL/SQL* is appropriate individuals who have experience in PL/SQL programming. and covers coding efficiency topics, object-oriented programming, working with external code, and the advanced features of Oracle-supplied packages.

**Curriculum Map-3**

**Languages Curriculum for Oracle9*i***

**Introduction to Oracle9*i*: SQL**

**or**

**Introduction to Oracle9*i* for**

**Experienced SQL Users**

**SQL for End Users**

**Extended Data inClass Retrieval**

**SQL1 with SQL inClass**

**inClass**

**Introduction to Oracle9*i*: PL/SQL**

**Develop PL/SQL**

**Program Units PL/SQL**

**Fundamentals**

**inClass**

**Advanced PL/SQL**

**inClass**

Copyright © Oracle Corporation, 2001. All rights reserved.

**Integrated Languages Curriculum**

The slide lists various modules and courses that are available in the languages curriculum. The following table lists the modules and courses with their equivalent TBTs.

| **Course or Module** | **Equivalent TBT** |
| --- | --- |
| *SQL1* | *Oracle SQL: Basic SELECT Statements*  *Oracle SQL: Data Retrieval Techniques*  *Oracle SQL: DML and DDL* |
| *Extended Data Retrieval with SQL* | *Oracle SQL and SQL\*Plus: Advanced SELECT Statements Oracle SQL and SQL\*Plus: SQL\*Plus and Reporting* |
| *Introduction to Oracle9i for Experienced SQL Users* | *Oracle SQL Specifics: Retrieving and Formatting Data Oracle SQL Specifics: Creating and Managing Database Objects* |
| *PL/SQL Fundamentals* | *PL/SQL: Basics* |
| *Develop PL/SQL Program Units* | *PL/SQL: Procedures, Functions, and Packages*  *PL/SQL: Database Programming* |
| *SQL for End Users* | *SQL for End Users: Part 1*  *SQL for End Users: Part 2* |
| *Advanced PL/SQL* | *Advanced PL/SQL: Implementation and Advanced Features Advanced PL/SQL: Design Considerations and Object Types* |

**Curriculum Map-4**

**Introduction**

Copyright © Oracle Corporation, 2001. All rights reserved.

**Objectives**

**After completing this lesson, you should be able**

**to do the following:**

• **List the features of Oracle9*i***

• **Discuss the theoretical and physical aspects of**

**a relational database**

• **Describe the Oracle implementation of the**

**RDBMS and ORDBMS**

**I-2** Copyright © Oracle Corporation, 2001. All rights reserved.

**Lesson Aim**

In this lesson, you gain an understanding of the relational database management system (RDBMS) and the object relational database management system (ORDBMS). You are also introduced to the following:

• SQL statements that are specific to the Oracle Server

• *i*SQL\*Plus, which is used for executing SQL and for formatting and reporting purposes

**Introduction to Oracle9*i:* SQL I-2**

**Oracle9*i***

**Scalability**

**Reliability**

**One**

**vendor**

**One mgmt. interface**

**Single dev. model**

**Common skill sets**

**I-3** Copyright © Oracle Corporation, 2001. All rights reserved.

**Oracle9*i* Features**

Oracle offers a comprehensive high-performance infrastructure for e-business. It is called Oracle9*i*. Oracle9*i* includes everything needed to develop, deploy, and manage Internet applications.

Benefits include:

• Scalability from departments to enterprise e-business sites

• Robust reliable, available, secure architecture

• One development model, easy deployment options

• Leverage an organization’s current skillset throughout the Oracle platform (including SQL, PL/SQL, Java, and XML)

• One management interface for all applications

• Industry standard technologies, no proprietary lock-in

**Introduction to Oracle9*i:* SQL I-3**

**Oracle9*i***

**I-4** Copyright © Oracle Corporation, 2001. All rights reserved.

**Oracle9*i***

There are two products, Oracle9*i* Application Server and Oracle9*i* Database, that provide a complete and simple infrastructure for Internet applications.

**Introduction to Oracle9*i:* SQL I-4**

**Oracle9*i* Application Server**

**Portals**

A

**Transactional Apps**

P

A

C

**Business Intelligence Business intelligence**

H

E

**Integration**

****

**I-5** Copyright © Oracle Corporation, 2001. All rights reserved.

**Oracle9*i* Application Server**

The Oracle9*i* Application Server (Oracle9*i*AS) runs all your applications. The Oracle9*i* Database stores all your data.

Oracle9*i* Application Server is the only application server to include services for all the different server applications you’ll want to run. Oracle9*i*AS can run your:

• Portals or Web sites

• Java transactional applications

• Business intelligence applications

It also provides integration between users, applications, and data throughout your organization.

**Introduction to Oracle9*i:* SQL I-5**

**Oracle9*i* Database Object Relational Data**

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Documents**

**XML**

**Multimedia**

**Messages**

****

**I-6** Copyright © Oracle Corporation, 2001. All rights reserved.

**Oracle9*i* Database**

The roles of the two products are very straightforward. Oracle9*i* Database manages all your data. This is not just the object relational data that you expect an enterprise database to manage. It can also be unstructured data like:

• Spreadsheets

• Word documents

• Powerpoint presentations

• XML

• Multimedia data types like MP3, graphics, video, and more

The data does not even have to be in the database. Oracle9*i* Database has services through which you can store metadata about information stored in file systems. You can use the database server to manage and serve information wherever it is located.

**Introduction to Oracle9*i:* SQL I-6**

**Oracle9*i* Database**

• **Performance and availability leader**

• **Richest feature set**

**I-7** Copyright © Oracle Corporation, 2001. All rights reserved.

**Oracle9*i* Database**

The starting point for any discussion about application deployment is the database. Oracle9*i* Database is the new flagship product from Oracle. It has an incredibly rich feature set.

Oracle9*i* Database is the only database specifically designed as an Internet development and deployment platform, extending Oracle's long-standing technology leadership in the areas of data management, transaction processing, and data warehousing to the new medium of the Internet. Built directly inside the database, breakthrough Internet features help companies and developers build Internet-savvy applications that lower costs, enhance customer and supplier interaction, and provide global information access across platforms and across the enterprise.

The Oracle9*i* Database is an object relational database management system. It has the full capabilities and functionality of a relational database, plus the features of an object database.

**Introduction to Oracle9*i:* SQL I-7**

**Oracle9*i*: Object Relational Database**

**Management System**

• **User-defined data types and objects**

• **Fully compatible with relational database**

• **Support of multimedia and large objects**

• **High-quality database server features**

**I-8** Copyright © Oracle Corporation, 2001. All rights reserved.

**About Oracle9*i***

The Oracle server extends the data modeling capabilities to support an object relational database model that brings object-oriented programming, complex data types, complex business objects, and full compatibility with the relational world.

It includes several features for improved performance and functionality of online transaction processing (OLTP) applications, such as better sharing of run-time data structures, larger buffer caches, and deferrable constraints. Data warehouse applications will benefit from enhancements such as parallel execution of in sert, update, and delete operations; partitioning; and parallel-aware query optimization. Operating within the Network Computing Architecture (NCA) framework, Oracle9*i* supports client-server and Web-based applications that are distributed and multitiered.

Oracle9*i* can scale tens of thousands of concurrent users, support up to 512 petabytes of data (a pedabyte is 1,000 terabytes), and can handle any type of data, including text, spatial, image, sound, video, and time series as well as traditional structured data.

For more information, see *Oracle9i Concepts*.

**Introduction to Oracle9*i:* SQL I-8**

**Oracle Internet Platform**

**Clients**

**Any mail**

**t**

**n**

**e**

**m**

**e**

**Any browser Any FTP client client**

****

**Internet applications**

**D**

**e**

**v**

**e**

**lo**

**SQL**

**ga**

**n**

**a**

**m**

***Business logic and data***

***Presentation and business logic***

**pm**

**e**

**n**

**t**

**PL/SQL**

**m**

**e**

**t**

**s**

**yS**

**Databases Application servers**

**Network services**

**to**

**o**

**ls**

****

**Java**

**I-9** Copyright © Oracle Corporation, 2001. All rights reserved.

**Oracle Internet Platform**

Oracle Corporation offers a comprehensive high-performance Internet platform for e-commerce and data warehousing. This integrated platform includes everything needed to develop, deploy, and manage Internet applications. The Oracle Internet Platform is built on three core pieces:

• Browser-based clients to process presentation

• Application servers to execute business logic and serve presentation logic to browser-based clients • Databases to execute database-intensive business logic and serve data

Oracle Corporation offers a wide variety of the most advanced graphical user interface (GUI) driven development tools to build business applications, as well as a large suite of software applications for many areas of business and industry. Stored procedures, functions, and packages can be written by using SQL, PL/SQL, or Java.

**Introduction to Oracle9*i:* SQL I-9**

**System Development Life Cycle**

**Strategy**

**and**

**analysis**

**Design**

**Build**

**and**

**document**

**Transition**

**Production**

**I-10** Copyright © Oracle Corporation, 2001. All rights reserved.

**System Development Life Cycle**

From concept to production, you can develop a database by using the system development life cycle, which contains multiple stages of development. This top-down, systematic approach to database development transforms business information requirements into an operational database.

**Strategy and Analysis**

• Study and analyze the business requirements. Interview users and managers to identify the information requirements. Incorporate the enterprise and application mission statements as well as any future system specifications.

• Build models of the system. Transfer the business narrative into a graphical representation of business information needs and rules. Confirm and refine the model with the analysts and experts.

**Design**

Design the database based on the model developed in the strategy and analysis phase. **Build and Document**

• Build the prototype system. Write and execute the commands to create the tables and supporting objects for the database.

• Develop user documentation, Help text, and operations manuals to support the use and operation of the system.

**Introduction to Oracle9*i:* SQL I-10**

**System Development Life Cycle**

**Strategy**

**and**

**analysis**

**Design**

**Build**

**and**

**document**

**Transition**

**Production**

**I-11** Copyright © Oracle Corporation, 2001. All rights reserved.

**System Development Life Cycle (continued)**

**Transition**

Refine the prototype. Move an application into production with user acceptance testing, conversion of existing data, and parallel operations. Make any modifications required.

**Production**

Roll out the system to the users. Operate the production system. Monitor its performance, and enhance and refine the system.

**Note:** The various phases of the system development life cycle can be carried out iteratively. This course focuses on the build phase of the system development life cycle.

**Introduction to Oracle9*i:* SQL I-11**

**Data Storage on Different Media**

****

**Database**

**Electronic spreadsheet**

**Filing cabinet**

**I-12** Copyright © Oracle Corporation, 2001. All rights reserved.

**Storing Information**

Every organization has some information needs. A library keeps a list of members, books, due dates, and fines. A company needs to save information about employees, departments, and salaries. These pieces of information are called *data****.***

Organizations can store data on various media and in different formats— for example, a hard-copy document in a filing cabinet or data stored in electronic spreadsheets or in databases.

A *database* is an organized collection of information.

To manage databases, you need database management systems (DBMS). A DBMS is a program that stores, retrieves, and modifies data in the database on request. There are four main types of databases: *hierarchical*, *network*, *relational*, and more recently *object relational****.***

**Note:** Oracle7 is a relational database management system and Oracle8, 8*i*, and 9*i* are object relational database management systems.

**Introduction to Oracle9*i:* SQL I-12**

**Relational Database Concept**

• **Dr. E.F. Codd proposed the relational model for**

**database systems in 1970.**

• **It is the basis for the relational database**

**management system.**

• **The relational model consists of the following:**

– **Collection of objects or relations**

– **Set of operators to act on the relations**

– **Data integrity for accuracy and consistency**

**I-13** Copyright © Oracle Corporation, 2001. All rights reserved.

**Relational Model**

The principles of the relational model were first outlined by Dr. E. F. Codd in a June 1970 paper called “A Relational Model of Data for Large Shared Data Banks.” In this paper, Dr. Codd proposed the relational model for database systems.

The more popular models used at that time were hierarchical and network, or even simple flat file data structures. Relational database management systems (RDBMS) soon became very popular, especially for their ease of use and flexibility in structure. In addition, a number of innovative vendors, such as Oracle, supplemented the RDBMS with a suite of powerful application development and user products, providing a total solution.

**Components of the Relational Model**

• Collections of objects or relations that store the data

• A set of operators that can act on the relations to produce other relations

• Data integrity for accuracy and consistency

For more information, see E. F. Codd*, The Relational Model for Database Management, Version 2* (Reading, Mass.: Addison-Wesley, 1990).

**Introduction to Oracle9*i:* SQL I-13**

**Definition of a Relational Database**

**A relational database is a collection of relations or**

**two-dimensional tables.**

**Oracle**

**server**

Table Name: **EMPLOYEES** Table Name: **DEPARTMENTS**

**I-14** Copyright © Oracle Corporation, 2001. All rights reserved.

**Definition of a Relational Database**

A relational database uses relations or two-dimensional tables to store information.

For example, you might want to store information about all the employees in your company. In a relational database, you create several tables to store different pieces of information about your employees, such as an employee table, a department table, and a salary table.

**Introduction to Oracle9*i:* SQL I-14**

**Data Models**

**Model of**

**system**

**in client’s mind**

**Entity model of**

**client’s model**

**Table model**

**of entity model**

**Oracle**

**~~server~~**

**Tables on disk**

**I-15** Copyright © Oracle Corporation, 2001. All rights reserved.

**Data Models**

Models are a cornerstone of design. Engineers build a model of a car to work out any details before putting it into production. In the same manner, system designers develop models to explore ideas and improve the understanding of the database design.

**Purpose of Models**

Models help communicate the concepts in people’s minds. They can be used to do the following: • Communicate

• Categorize

• Describe

• Specify

• Investigate

• Evolve

• Analyze

• Imitate

The objective is to produce a model that fits a multitude of these uses, can be understood by an end user, and contains sufficient detail for a developer to build a database system.

**Introduction to Oracle9*i:* SQL I-15**

**Entity Relationship Model**

• **Create an entity relationship diagram from business specifications or narratives**

**EMPLOYEE**

**#\* number \* name**

**o job title**

• **Scenario**

**assigned to**

**composed of**

**DEPARTMENT #\* number \* name o location**

– **“. . . Assign one or more employees to a**

**department . . .”**

– **“. . . Some departments do not yet have**

**assigned employees . . .”**

**I-16** Copyright © Oracle Corporation, 2001. All rights reserved.

**ER Modeling**

In an effective system, data is divided into discrete categories or entities. An entity relationship (ER) model is an illustration of various entities in a business and the rel ationships between them. An ER model is derived from business specifications or narratives and built during the analysis phase of the system development life cycle. ER models separate the information required by a business from the activities performed within a business. Although businesses can change their activities, the type of information tends to remain constant. Therefore, the data structures also tend to be constant.

**Benefits of ER Modeling**

• Documents information for the organization in a clear, precise format

• Provides a clear picture of the scope of the information requirement

• Provides an easily understood pictorial map for the database design

• Offers an effective framework for integrating multiple applications

**Key Components**

• Entity: A thing of significance about which information needs to be known. Examples are departments, employees, and orders.

• Attribute: Something that describes or qualifies an entity. For example, for the employee entity, the attributes would be the employee number, name, job title, hire date, department number, and so on. Each of the attributes is either required or optional. This state is called *optionality*.

• Relationship: A named association between entities showing optionality and degree. Examples are employees and departments, and orders and items.

**Introduction to Oracle9*i:* SQL I-16**

**Entity**

**Soft box**

**Entity Relationship Modeling Conventions**

**Attribute**

**Singular name**

**Singular, unique name Uppercase**

**Synonym in parentheses**

**EMPLOYEE**

**#\* number**

**assigned to**

**Lowercase**

**Mandatory marked with “\*” Optional marked with “o”**

**DEPARTMENT**

**#\* number**

**\* name o job title**

**composed of**

**\* name o location**

**Unique Identifier (UID)**

**Primary marked with “#”**

**Secondary marked with “(#)”**

**I-17** Copyright © Oracle Corporation, 2001. All rights reserved.

**ER Modeling (continued)**

**Entities**

To represent an entity in a model, use the following conventions:

• Soft box with any dimensions

• Singular, unique entity name

• Entity name in uppercase

• Optional synonym names in uppercase within parentheses: ( )

**Attributes**

To represent an attribute in a model, use the following conventions:

• Use singular names in lowercase.

• Tag mandatory attributes, or values that must be known, with an asterisk: \*. • Tag optional attributes, or values that may be known, with the letter o. **Relationships**

| **Symbol** | **Description** |
| --- | --- |
| Dashed line | Optional element indicating “may be” |
| Solid line | Mandatory element indicating “must be” |
| Crow’s foot | Degree element indicating “one or more” |
| Single line | Degree element indicating “one and only one” |

**Introduction to Oracle9*i:* SQL I-17**

**Entity**

**Soft box**

**Entity Relationship Modeling Conventions**

**Attribute**

**Singular name**

**Singular, unique name Uppercase**

**Synonym in parentheses**

**EMPLOYEE**

**#\* number**

**assigned to**

**Lowercase**

**Mandatory marked with “\*” Optional marked with “o”**

**DEPARTMENT**

**#\* number**

**\* name o job title**

**composed of**

**\* name o location**

**Unique Identifier (UID)**

**Primary marked with “#”**

**Secondary marked with “(#)”**

**I-18** Copyright © Oracle Corporation, 2001. All rights reserved.

**ER Modeling (continued)**

**Relationships**

Each direction of the relationship contains:

• A label, for example, *taught by* or *assigned to*

• An optionality, either *must be* or *may be*

• A degree, either *one and only one* or *one or more*

**Note:** The term *cardinality* is a synonym for the term *degree*.

Each source entity {may be | must be} relationship name {one and only one | one or more} destination entity.

**Note:** The convention is to read clockwise.

**Unique Identifiers**

A unique identifier (UID) is any combination of attributes or relationships, or both, that serves to distinguish occurrences of an entity. Each entity occurrence must be uniquely identifiable.

• Tag each attribute that is part of the UID with a number symbol: #

• Tag secondary UIDs with a number sign in parentheses: (#)

**Introduction to Oracle9*i:* SQL I-18**

**Relating Multiple Tables**

• **Each row of data in a table is uniquely**

**identified by a primary key (PK).**

• **You can logically relate data from multiple**

**tables using foreign keys (FK).**

Table Name: **DEPARTMENTS**

Table Name: **EMPLOYEES Primary key Foreign key Primary key**

**I-19** Copyright © Oracle Corporation, 2001. All rights reserved.

**Relating Multiple Tables**

Each table contains data that describes exactly one entity. For example, the EMPLOYEES table contains information about employees. Categories of data are listed across the top of each table, and individual cases are listed below. Using a table format, you can readily visualize, understand, and use information.

Because data about different entities is stored in different tables, you may need to combine two or more tables to answer a particular question. For example, you may want to know the location of the department where an employee works. In this scenario, you need information from the EMPLOYEES table (which contains data about employees) and the DEPARTMENTS table (which contains information about departments). With an RDBMS you can relate the data in one table to the data in another by using the foreign keys. A foreign key is a column or a set of columns that refer to a primary key in the same table or another table.

You can use the ability to relate data in one table to data in another to organize information in separate, manageable units. Employee data can be kept logically distinct from department data by storing it in a separate table.

**Guidelines for Primary Keys and Foreign Keys**

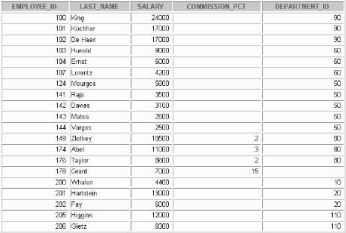
• You can not use duplicate values in a primary key.

• Primary keys generally cannot be changed.

• Foreign keys are based on data values and are purely logical, not physical, pointers. • A foreign key value must match an existing primary key value or unique key value, or else be null. • A foreign key must reference either a primary key or unique key column.

**Introduction to Oracle9*i:* SQL I-19**

**Relational Database Terminology 2 3 4**

**6** 

**5**

**1**

**I-20** Copyright © Oracle Corporation, 2001. All rights reserved.

**Terminology Used in a Relational Database**

A relational database can contain one or many tables. A table is the basic storage structure of an RDBMS. A table holds all the data necessary about something in the real world: for example, employees, invoices, or customers.

The slide shows the contents of the EMPLOYEES table or *relation.* The numbers indicate the following: 1. A single row or tuple representing all data required for a particular employee. Each row in a table should be identified by a primary key, which allows no duplicate rows. The order of rows is insignificant; specify the row order when the data is retrieved.

2. A column or attribute containing the employee number. The employee number identifies a unique employee in the EMPLOYEES table. In this example, the employee number column is designated as the *primary key*. A primary key must contain a value, and the value must be unique.

3. A column that is not a key value. A column represents one kind of data in a table; in the example, the salary of all the employees. Column order is insignificant when storing data; specify the column order when the data is retrieved.

4. A column containing the department number, which is also a foreign key. A *foreign key* is a column that defines how tables relate to each other. A foreign key refers to a primary key or a unique key in in the same table or in another table. In the example, DEPARTMENT\_ID *uniquely* identifies a department in the DEPARTMENTS table.

5. A field may have no value in it. This is called a null value. In the EMPLOYEES table, only employees who have a role of sales representative have a value in the COMMISSION\_PCT(commission) field.

6. A field can be found at the intersection of a row and a column. There can be only one value in it.

**Introduction to Oracle9*i:* SQL I-20**

**Relational Database Properties**

**A relational database:**

• **Can be accessed and modified by executing**

**structured query language (SQL) statements**

• **Contains a collection of tables with no physical**

**pointers**

• **Uses a set of operators**

**I-21** Copyright © Oracle Corporation, 2001. All rights reserved.

**Properties of a Relational Database**

In a relational database, you do not specify the access route to the tables, and you do not need to know how the data is arranged physically.

To access the database, you execute a structured query language (SQL) statement, which is the American National Standards Institute (ANSI) standard language for operating relational databases. The language contains a large set of operators for partitioning and combining relations. The database can be modified by using SQL statements.

**Introduction to Oracle9*i:* SQL I-21**

**Communicating with a RDBMS Using SQL**

**SQL statement**

**is entered.**

**SELECT department\_name FROM departments;**

**Data is displayed.**

****

**Statement is sent to Oracle Server.**

**Oracle**

**server**

**I-22** Copyright © Oracle Corporation, 2001. All rights reserved.

**Structured Query Language**

Using SQL, you can communicate with the Oracle server. SQL has the following advantages: • Efficient

• Easy to learn and use

• Functionally complete (With SQL, you can define, retrieve, and manipulate data in the tables.)

**Introduction to Oracle9*i:* SQL I-22**

**Relational Database Management System**

**Oracle**

**server**

**User tables Data**

**dictionary**

**I-23** Copyright © Oracle Corporation, 2001. All rights reserved.

**Relational Database Management System**

Oracle provides a flexible RDBMS called Oracle9*i*. Using its features, you can store and manage data with all the advantages of a relational structure plus PL/SQL, an engine that provides you with the ability to store and execute program units. Oracle9*i* also supports Java and XML. The Oracle server offers the options of retrieving data based on optimization techniques. It includes security features that control how a database is accessed and used. Other features include consistency and protection of data through locking mechanisms.

The Oracle9*i* server is an object-relational database management system that provides an open, comprehensive, and integrated approach to information management. An Oracle server consists of an Oracle database and an Oracle server instance. Every time a database is started, a system global area (SGA) is allocated, and Oracle background processes are started. The system global area is an area of memory used for database information shared by the database users. The combination of the background processes and memory buffers is called an Oracle instance.

**Introduction to Oracle9*i:* SQL I-23**

**SELECT**

**INSERT**

**UPDATE**

**DELETE**

**MERGE**

**CREATE**

**ALTER**

**DROP**

**RENAME**

**TRUNCATE**

**COMMIT**

**ROLLBACK SAVEPOINT**

**GRANT**

**REVOKE**

**SQL Statements**

**Data retrieval**

**Data manipulation language (DML) Data definition language (DDL)**

**Transaction control**

**Data control language (DCL)**

**I-24** Copyright © Oracle Corporation, 2001. All rights reserved.

**SQL Statements**

Oracle SQL complies with industry-accepted standards. Oracle Corporation ensures future compliance with evolving standards by actively involving key personnel in SQL standards committees. Industry accepted committees are the American National Standards Institute (ANSI) and the International Standards Organization (ISO). Both ANSI and ISO have accepted SQL as the standard language for relational databases.

| **Statement** | **Description** |
| --- | --- |
| SELECT | Retrieves data from the database |
| INSERT  UPDATE  DELETE  MERGE | Enters new rows, changes existing rows, and removes unwanted rows from tables in the database, respectively. Collectively known as *data manipulation language* (DML). |
| CREATE  ALTER  DROP  RENAME  TRUNCATE | Set up, change, and remove data structures from tables. Collectively known as *data definition language* (DDL). |
| COMMIT  ROLLBACK  SAVEPOINT | Manage the changes made by DML statements. Changes to the data can be grouped together into logical transactions. |
| GRANT  REVOKE | Give or remove access rights to both the Oracle database and the structures within it. Collectively known as *data control language* (DCL). |

**Introduction to Oracle9*i:* SQL I-24**

**Tables Used in the Course**

**EMPLOYEES**

**DEPARTMENTS JOB\_GRADES**

**I-25** Copyright © Oracle Corporation, 2001. All rights reserved.

**Tables Used in the Course**

The following main tables are used in this course:

• EMPLOYEES table, which gives details of all the employees

• DEPARTMENTS table, which gives details of all the departments

• JOB\_GRADES table, which gives details of salaries for various grades

**Note:** The structure and data for all the tables are provided in Appendix B.

**Introduction to Oracle9*i:* SQL I-25**

**Summary**

• **The Oracle9*i* Server is the database for Internet**

**computing.**

• **Oracle9*i* is based on the object relational database management system.**

• **Relational databases are composed of relations, managed by relational operations, and governed**

**by data integrity constraints.**

• **With the Oracle Server, you can store and manage information by using the SQL language and**

**PL/SQL engine.**

**I-26** Copyright © Oracle Corporation, 2001. All rights reserved.

**Summary**

Relational database management systems are composed of objects or relations. They are managed by operations and governed by data integrity constraints.

Oracle Corporation produces products and services to meet your relational database management system needs. The main products are the Oracle9*i* Database Server, with which you can store and manage information by using SQL, and the Oracle9*i* Application Server with which you can run all of your applications.

**SQL**

The Oracle Server supports ANSI standard SQL and contains extensions. SQL is the language used to communicate with the server to access, manipulate, and control data**.**

**Introduction to Oracle9*i:* SQL I-26**

**Writing Basic**

**SQL SELECT Statements**

****Copyright © Oracle Corporation, 2001. All rights reserved.

**Objectives**

**After completing this lesson, you should be able to**

**do the following:**

• **List the capabilities of SQL SELECT statements**

• **Execute a basic SELECT statement**

• **Differentiate between SQL statements and**

***i*SQL\*Plus commands**

**1-2** Copyright © Oracle Corporation, 2001. All rights reserved.

**Lesson Aim**

To extract data from the database, you need to use the structured query language (SQL) SELECT statement. You may need to restrict the columns that are displayed. This lesson describes all the SQL statements that you need to perform these actions.

You may want to create SELECT statements that can be used more than once. This lesson also covers the *i*SQL\*Plus environment where you execute SQL statements.

**Note:** *i*SQL\*Plus is new in the Oracle9*i* product. It is a browser environment where you execute SQL commands. In pior releases of Oracle, SQL\*Plus was the default environment where you executed SQL commands. SQL\*Plus is still available and is described in Appendix C.

**Introduction to Oracle9*i:* SQL 1-2**

**Capabilities of SQL SELECT Statements**

**Projection Selection**

**Table 1 Table 1**

**Join**

**Table 1 Table 2**

**1-3** Copyright © Oracle Corporation, 2001. All rights reserved.

**Capabilities of SQL SELECT Statements**

A SELECT statement retrieves information from the database. Using a SELECT statement, you can do the following:

• Projection: You can use the projection capability in SQL to choose the columns in a table that you want returned by your query. You can choose as few or as many columns of the table as you require.

• Selection: You can use the selection capability in SQL to choose the rows in a table that you want returned by a query. You can use various criteria to restrict the rows that you see.

• Joining: You can use the join capability in SQL to bring together data that is stored in different tables by creating a link between them. You learn more about joins in a later lesson.

**Introduction to Oracle9*i:* SQL 1-3**

**Basic SELECT Statement**

**SELECT \*|{[DISTINCT] *column*|*expression* [*alias*],...} FROM *table;***

• **SELECT identifies *what* columns**

• **FROM identifies *which* table**

**1-4** Copyright © Oracle Corporation, 2001. All rights reserved.

**Basic SELECT Statement**

In its simplest form, a SELECT statement must include the following:

• A SELECT clause, which specifies the columns to be displayed

• A FROM clause, which specifies the table containing the columns listed in the SELECT clause In the syntax:

SELECT is a list of one or more columns

\* selects all columns

DISTINCT suppresses duplicates

*column|expression* selects the named column or the expression

*alias* gives selected columns different headings

FROM *table* specifies the table containing the columns

**Note:** Throughout this course, the words *keyword*, *clause*, and *statement* are used as follows.

• A *keyword* refers to an individual SQL element.

For example, SELECT and FROM are keywords.

• A *clause* is a part of a SQL statement.

For example, SELECT employee\_id, last\_name, ... is a clause.

• A *statement* is a combination of two or more clauses.

For example, SELECT \* FROM employees is a SQL statement.

**Introduction to Oracle9*i:* SQL 1-4**

**Selecting All Columns**

**SELECT \***

**FROM departments;**

****

**1-5** Copyright © Oracle Corporation, 2001. All rights reserved.

**Selecting All Columns of All Rows**

You can display all columns of data in a table by following the SELECT keyword with an asterisk (\*). In the example on the slide, the department table contains four columns: DEPARTMENT\_ID, DEPARTMENT\_NAME, MANAGER\_ID, and LOCATION\_ID. The table contains seven rows, one for each department.

You can also display all columns in the table by listing all the columns after the SELECT keyword. For example, the following SQL statement, like the example on the slide, displays all columns and all rows of the DEPARTMENTS table:

SELECT department\_id, department\_name, manager\_id, location\_ id FROM departments;

**Introduction to Oracle9*i:* SQL 1-5**

**Selecting Specific Columns**

**SELECT department\_id, location\_id**

**FROM departments;**

****

**1-6** Copyright © Oracle Corporation, 2001. All rights reserved.

**Selecting Specific Columns of All Rows**

You can use the SELECT statement to display specific columns of the table by specifying the column names, separated by commas. The example on the slide displays all the department numbers and location numbers from the DEPARTMENTS table.

In the SELECT clause, specify the columns that you want, in the order in whic h you want them to appear in the output. For example, to display location before department number going from left to right, you use the following statement:

SELECT location\_id, department\_id

FROM departments;



**Introduction to Oracle9*i:* SQL 1-6**

**Writing SQL Statements**

• **SQL statements are not case sensitive.**

• **SQL statements can be on one or more lines.**

• **Keywords cannot be abbreviated or split**

**across lines.**

• **Clauses are usually placed on separate lines.**

• **Indents are used to enhance readability.**

**1-7** Copyright © Oracle Corporation, 2001. All rights reserved.

**Writing SQL Statements**

Using the following simple rules and guidelines, you can construct valid statements that are both easy to read and easy to edit:

• SQL statements are not case sensitive, unless indicated.

• SQL statements can be entered on one or many lines.

• Keywords cannot be split across lines or abbreviated.

• Clauses are usually placed on separate lines for readability and ease of editing. • Indents should be used to make code more readable.

• Keywords typically are entered in uppercase; all other words, such as table names and columns, are entered in lowercase.

**Executing SQL Statements**

Using *i*SQL\*Plus, click the Execute button to run the command or commands in the editing window.

**Introduction to Oracle9*i:* SQL 1-7**

**Column Heading Defaults**

• ***i*SQL\*Plus:**

– **Default heading justification: Center**

– **Default heading display: Uppercase**

• **SQL\*Plus:**

– **Character and Date column headings are left**

**justified**

– **Number column headings are right-justified**

– **Default heading display: Uppercase**

**1-8** Copyright © Oracle Corporation, 2001. All rights reserved.

**Column Heading Defaults**

In *i*SQL\*Plus, column headings are displayed in uppercase and centered.

SELECT last\_name, hire\_date, salary

FROM employees;

You can override the column heading display with an alias. Column aliases are covered later in this lesson.

**Introduction to Oracle9*i:* SQL 1-8**

**Arithmetic Expressions**

**Create expressions with number and date data by using arithmetic operators.**

**Operator +**

**-**

**\***

**/**

**Description Add**

**Subtract Multiply**

**Divide**

**1-9** Copyright © Oracle Corporation, 2001. All rights reserved.

**Arithmetic Expressions**

You may need to modify the way in which data is displayed, perform calculations, or look at what-if scenarios. These are all possible using arithmetic expressions. An arithmetic expression can contain column names, constant numeric values, and the arithmetic operators.

**Arithmetic Operators**

The slide lists the arithmetic operators available in SQL. You can use arithmetic operators in any clause of a SQL statement except in the FROM clause.

**Introduction to Oracle9*i:* SQL 1-9**

**Using Arithmetic Operators**

**SELECT last\_name, salary, salary + 300**

**FROM employees;**

****

**1-10** Copyright © Oracle Corporation, 2001. All rights reserved.

**Using Arithmetic Operators**

The example in the slide uses the addition operator to calculate a salary increase of $300 for all employees and displays a new SALARY+300 column in the output.

Note that the resultant calculated column SALARY+300 is not a new column in the EMPLOYEES table; it is for display only. By default, the name of a new column comes from the calculation that generated it— in this case, salary+300.

**Note:** The Oracle9*i* server ignores blank spaces before and after the arithmetic operator.

**Introduction to Oracle9*i:* SQL 1-10**

**Operator Precedence**

**\*/ +\_**

• **Multiplication and division take priority over**

**addition and subtraction.**

• **Operators of the same priority are evaluated from left to right.**

• **Parentheses are used to force prioritized**

**evaluation and to clarify statements.**

**1-11** Copyright © Oracle Corporation, 2001. All rights reserved.

**Operator Precedence**

If an arithmetic expression contains more than one operator, multiplication and division are evaluated first. If operators within an expression are of same priority, then evaluation is done from left to right.

You can use parentheses to force the expression within parentheses to be evaluated first.

**Introduction to Oracle9*i:* SQL 1-11**

**Operator Precedence**

**SELECT last\_name, salary, 12\*salary+100**

**FROM employees;**

****...

**1-12** Copyright © Oracle Corporation, 2001. All rights reserved.

**Operator Precedence (continued)**

The example in the slide displays the last name, salary, and annual compensation of each employee. It calculates the annual compensation as 12 multiplied by the monthly salary, plus a one-time bonus of $100. Notice that multiplication is performed before addition.

**Note:** Use parentheses to reinforce the standard order of precedence and to improve clarity. For example, the expression on the slide can be written as (12\*salary)+100 with no change in the result.

**Introduction to Oracle9*i:* SQL 1-12**

**Using Parentheses**

**SELECT last\_name, salary, 12\*(salary+100)**

**FROM employees;**

****

**1-13** Copyright © Oracle Corporation, 2001. All rights reserved.

**Using Parentheses**

You can override the rules of precedence by using parentheses to specify the order in which operators are executed.

The example in the slide displays the last name, salary, and annual compensation of each employee. It calculates the annual compensation as monthly salary plus a monthly bonus of $100, multiplied by 12. Because of the parentheses, addition takes priority over multiplication.

**Introduction to Oracle9*i:* SQL 1-13**

**Defining a Null Value**

• **A null is a value that is unavailable, unassigned,**

**unknown, or inapplicable.**

• **A null is not the same as zero or a blank space.**

**SELECT last\_name, job\_id, salary, commission\_pct**

**FROM employees;**

****... 

**1-14** Copyright © Oracle Corporation, 2001. All rights reserved.

**Null Values**

If a row lacks the data value for a particular column, that value is said to be *null*, or to contain a null.

A null is a value that is unavailable, unassigned, unknown, or inapplicable. A null is not the same as zero or a space. Zero is a number, and a space is a character.

Columns of any data type can contain nulls. However, some constraints, NOT NULL and PRIMARY KEY, prevent nulls from being used in the column.

In the COMMISSION\_PCT column in the EMPLOYEES table, notice that only a sales manager or sales representative can earn a commission. Other employees are not entitled to earn commissions. A null represents that fact.

**Introduction to Oracle9*i:* SQL 1-14**

**Null Values**

**in Arithmetic Expressions**

**Arithmetic expressions containing a null value**

**evaluate to null.**

**SELECT last\_name, 12\*salary\*commission\_pct**

**FROM employees;**

****

**1-15** Copyright © Oracle Corporation, 2001. All rights reserved.

**Null Values (continued)**

If any column value in an arithmetic expression is null, the result is null. For example, if you attempt to perform division with zero, you get an error. However, if you divide a number by null, the result is a null or unknown.

In the example on the slide, employee King does not get any commission. Because the COMMISSION\_PCT column in the arithmetic expression is null, the result is null.

For more information, see *Oracle9i SQL Reference,* “Basic Elements of SQL.”

**Introduction to Oracle9*i:* SQL 1-15**

**Defining a Column Alias**

**A column alias:**

• **Renames a column heading**

• **Is useful with calculations**

• **Immediately follows the column name: there can also be the optional AS keyword between the**

**column name and alias**

• **Requires double quotation marks if it contains**

**spaces or special characters or is case sensitive**

**1-16** Copyright © Oracle Corporation, 2001. All rights reserved.

**Column Aliases**

When displaying the result of a query, *i*SQL\*Plus normally uses the name of the selected column as the column heading. This heading may not be descriptive and hence may be difficult to understand. You can change a column heading by using a column alias.

Specify the alias after the column in the SELECT list using a space as a separator. By default, alias headings appear in uppercase. If the alias contains spaces or special characters (such as # or $), or is case sensitive, enclose the alias in double quotation marks (" ").

**Introduction to Oracle9*i:* SQL 1-16**

**Using Column Aliases**

**SELECT last\_name AS name, commission\_pct comm**

**FROM employees;**

****

**SELECT last\_name "Name",**

**salary\*12 "Annual Salary"**

**FROM employees;**

**1-17** Copyright © Oracle Corporation, 2001. All rights reserved.

**Column Aliases (continued)**

The first example displays the names and the commission percentages of all the employees. Notice that the optional AS keyword has been used before the column alias name. The result of the query is the same whether the AS keyword is used or not. Also notice that the SQL statement has the column aliases, name and comm, in lowercase, whereas the result of the query displays the column headings in uppercase. As mentioned in a previous slide, column headings appear in uppercase by default.

The second example displays the last names and annual salaries of all the employees. Because Annual Salary contains a space, it has been enclosed in double quotation marks. Notice that the column heading in the output is exactly the same as the column alias.

**Introduction to Oracle9*i:* SQL 1-17**

**Concatenation Operator**

**A concatenation operator:**

• **Concatenates columns or character strings to**

**other columns**

• **Is represented by two vertical bars (||)**

• **Creates a resultant column that is a character**

**expression**

**1-18** Copyright © Oracle Corporation, 2001. All rights reserved.

**Concatenation Operator**

You can link columns to other columns, arithmetic expressions, or constant values to create a character expression by using the concatenation operator (||). Columns on either side of the operator are combined to make a single output column.

**Introduction to Oracle9*i:* SQL 1-18**

**Using the Concatenation Operator**

**SELECT last\_name||job\_id AS "Employees"**

**FROM employees;**

****

**1-19** Copyright © Oracle Corporation, 2001. All rights reserved.

**Using the Concatenation Operator**

In the example, LAST\_NAME and JOB\_ID are concatenated, and they are given the alias Employees. Notice that the employee last name and job code are combined to make a single output column. The AS keyword before the alias name makes the SELECT clause easier to read.

**Introduction to Oracle9*i:* SQL 1-19**

**Literal Character Strings**

• **A literal value is a character, a number, or a date included in the SELECT list.**

• **Date and character literal values must be enclosed within single quotation marks.**

• **Each character string is output once for each**

**row returned.**

**1-20** Copyright © Oracle Corporation, 2001. All rights reserved.

**Literal Character Strings**

A literal value is a character, a number, or a date that is included in the SELECT list and that is not a column name or a column alias. It is printed for each row returned. Literal strings of free-format text can be included in the query result and are treated the same as a column in the SELECT list.

Date and character literals *must* be enclosed within single quotation marks (' '); number literals need not.

**Introduction to Oracle9*i:* SQL 1-20**

**Using Literal Character Strings**

**SELECT last\_name ||' is a '||job\_id**

**AS "Employee Details"**

**FROM employees;**

****

**1-21** Copyright © Oracle Corporation, 2001. All rights reserved.

**Using Literal Character Strings**

The example on the slide displays last names and job codes of all employees. The column has the heading Employee Details. Notice the spaces between the single quotation marks in the SELECT statement. The spaces improve the readability of the output.

In the following example, the last name and salary for each employee are concatenated with a literal to give the returned rows more meaning.

SELECT last\_name ||': 1 Month salary = '||salary Monthly FROM employees;



**Introduction to Oracle9*i:* SQL 1-21**

**Duplicate Rows**

**The default display of queries is all rows, including duplicate rows.**

**SELECT department\_id**

**FROM employees;**

****

**1-22** Copyright © Oracle Corporation, 2001. All rights reserved.

**Duplicate Rows**

Unless you indicate otherwise, *i*SQL\*Plus displays the results of a query without eliminating duplic ate rows. The example on the slide displays all the department numbers from the EMPLOYEES table. Notice that the department numbers are repeated.

**Introduction to Oracle9*i:* SQL 1-22**