D17108GC30

Edition 3.0

January 2009

D57872



**Oracle Database 10*g*: SQL**

**Fundamentals I**

**Student Guide • Volume 3**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Authors**

Salome Clement Chaitanya Koratamaddi Nancy Greenberg

**Technical Contributors**

**and Reviewers**

Wayne Abbott

Christian Bauwens

Claire Bennett

Perry Benson

Brian Boxx

Zarko Cesljas

Dairy Chan

Laszlo Czinkoczki

Joel Goodman

Matthew Gregory

Sushma Jagannath

Yash Jain

Angelika Krupp

Isabelle Marchand

Malika Marghadi

Valli Pataballa

Narayanan Radhakrishnan

Bryan Roberts

Helen Robertson

Lata Shivaprasad

John Soltani

James Spiller

Priya Vennapusa

**Editors**

Arijit Ghosh

Raj Kumar

**Graphic Designer**

Rajiv Chandrabhanu

**Publisher**

Giri Venugopal

**Copyright © 2009, Oracle. All rights reserved.**

**Disclaimer**

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

**Restricted Rights Notice**

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government’s rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

**Trademark Notice**

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Contents**

**Preface**

1. **Introduction**

Lesson Objectives I-2 Goals of the Course I-3 Oracle10*g* I-4

Oracle Database 10*g* I-6

Oracle Application Server 10*g* I-7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Oracle Enterprise Manager 10*g* Grid Control | | | | I-8 |
| Relational and Object Relational Database Management Systems I-9 | | | | |
| Oracle Internet Platform | I-10 |  |  |  |
| System Development Life Cycle | | I-11 | |  |
| Data Storage on Different Media | | I-13 | |  |
| Relational Database Concept I-14 | | |  |  |
| Definition of a Relational Database | | | I-15 |  |
| Data Models I-16 |  |  |  |  |
| Entity Relationship Model | I-17 |  |  |  |
| Entity Relationship Modeling Conventions I-19 | | | | |
| Relating Multiple Tables | I-21 |  |  |  |
| Relational Database Terminology | | | I-23 |  |
| Relational Database Properties | | I-25 | |  |
| Communicating with an RDBMS Using SQL | | | | I-26 |

Oracle’s Relational Database Management System I-27

SQL Statements I-28

Tables Used in the Course I-29

Summary I-30

1. **Retrieving Data Using the SQL SELECT Statement** Objectives 1-2

Capabilities of SQL SELECT Statements 1-3 Basic SELECT Statement 1-4

Selecting All Columns 1-5 Selecting Specific Columns 1-6 Writing SQL Statements 1-7 Column Heading Defaults 1-8

**iii**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

Arithmetic Expressions 1-9

Using Arithmetic Operators 1-10

Operator Precedence 1-11

Defining a Null Value 1-12

Null Values in Arithmetic Expressions 1-13

Defining a Column Alias 1-14

Using Column Aliases 1-15

Concatenation Operator 1-16

Literal Character Strings 1-17

Using Literal Character Strings 1-18

Alternative Quote (q) Operator 1-19

Duplicate Rows 1-20

Development Environments for SQL 1-21

What Is Oracle SQL Developer? 1-22

Oracle SQL Developer Interface 1-23

Creating a Database Connection 1-24

Browsing Database Objects 1-27

Using the SQL Worksheet 1-28

Executing SQL Statements 1-31

Formatting the SQL Code 1-32

Saving SQL Statements 1-33

Running Script Files 1-34

Displaying the Table Structure 1-35

Using the DESCRIBE Command 1-36

Summary 1-37

Practice 1: Overview 1-38

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

Limiting Rows Using a Selection 2-3

Limiting the Rows That Are Selected 2-4

Using the WHERE Clause 2-5

Character Strings and Dates 2-6

Comparison Conditions 2-7

Using Comparison Conditions 2-8

Using the BETWEEN Condition 2-9

Using the IN Condition 2-10

Using the LIKE Condition 2-11

Using the NULL Conditions 2-13

Logical Conditions 2-14

**iv**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

Using the AND Operator 2-15

Using the OR Operator 2-16

Using the NOT Operator 2-17

Rules of Precedence 2-18

Using the ORDER BY Clause 2-20

Sorting 2-21

Substitution Variables 2-22

Using the & Substitution Variable 2-24

Character and Date Values with Substitution Variables 2- 26 Specifying Column Names, Expressions, and Text 2-27 Using the && Substitution Variable 2-28

Using the DEFINE Command 2-29

Using the VERIFY Command 2-30

Summary 2-31

Practice 2: Overview 2-32

1. **Using Single-Row Functions to Customize Output** 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 3-20

Using Arithmetic Operators with Dates 3-21 Date Functions 3-22

|  |  |
| --- | --- |
| Using Date Functions 3-23 |  |
| Practice 3: Overview of Part 1 | 3-25 |
| Conversion Functions 3-26 |  |
| Implicit Data Type Conversion | 3-27 |
| Explicit Data Type Conversion | 3-29 |

Using the TO\_CHAR Function with Dates 3-32

**v**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

Elements of the Date Format Model 3-33 Using the TO\_CHAR Function with Dates 3-37 Using the TO\_CHAR Function with Numbers 3-38 Using the TO\_NUMBER and TO\_DATE Functions 3-41

1. Date Format 3-43
2. Date Format: Example 3-44 Nesting Functions 3-45 General Functions 3-47

NVL Function 3-48

Using the NVL Function 3-49 Using the NVL2 Function 3-50 Using the NULLIF Function 3-51 Using the COALESCE Function 3-52 Conditional Expressions 3-54 CASE Expression 3-55

Using the CASE Expression 3-56 DECODE Function 3-57

Using the DECODE Function 3-58 Summary 3-60

Practice 3: Overview of Part 2 3-61

1. **Reporting Aggregated Data Using the Group Functions** Objectives 4-2

What Are Group Functions? 4-3 Types of Group Functions 4-4 Group Functions: Syntax 4-5

Using the AVG and SUM Functions 4-6 Using the MIN and MAX Functions 4-7 Using the COUNT Function 4-8 Using the DISTINCT Keyword 4-9 Group Functions and Null Values 4-10 Creating Groups of Data 4-11

Creating Groups of Data: GROUP BY Clause Syntax 4-12

|  |  |  |
| --- | --- | --- |
| Using the GROUP BY Clause 4-13 |  |  |
| Grouping by More Than One Column | 4-15 |  |
| Using the GROUP BY Clause on Multiple Columns | | 4-16 |
| Illegal Queries Using Group Functions | 4-17 |  |
| Restricting Group Results 4-19 |  |  |
| Restricting Group Results with the HAVING Clause | | 4-20 |

**vi**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

Using the HAVING Clause 4-21

Nesting Group Functions 4-23

Summary 4-24

Practice 4: Overview 4-25

1. **Displaying Data from Multiple Tables** Objectives 5-2

Obtaining Data from Multiple Tables 5-3 Types of Joins 5-4

Joining Tables Using SQL:1999 Syntax 5-5 Creating Natural Joins 5-6

|  |  |
| --- | --- |
| Retrieving Records with Natural Joins | 5-7 |
| Creating Joins with the USING Clause | 5-8 |

Joining Column Names 5-9

Retrieving Records with the USING Clause 5-10

|  |  |
| --- | --- |
| Qualifying Ambiguous Column Names | 5-11 |
| Using Table Aliases 5-12 |  |
| Creating Joins with the ON Clause 5-13 | |
| Retrieving Records with the ON Clause | 5-14 |

Self-Joins Using the ON Clause 5-15 Applying Additional Conditions to a Join 5-17 Creating Three-Way Joins with the ON Clause 5-18

|  |  |  |  |
| --- | --- | --- | --- |
| Nonequijoins 5-19 | |  |  |
| Retrieving Records with Nonequijoins | | | 5-20 |
| Outer Joins 5-21 |  |  |  |
| INNER Versus OUTER Joins 5-22 | | |  |
| LEFT OUTER JOIN |  | 5-23 |  |
| RIGHT OUTER JOIN | | 5-24 |  |
| FULL OUTER JOIN |  | 5-25 |  |
| Cartesian Products | | 5-26 |  |
| Generating a Cartesian Product 5-27 | | |  |
| Creating Cross Joins 5-28 | | |  |
| Summary 5-29 |  |  |  |
| Practice 5: Overview 5-30 | | |  |
| **6 Using Subqueries to Solve Queries** | | |  |
| Objectives 6-2 |  |  |  |
| Using a Subquery to Solve a Problem | | | 6-3 |
| Subquery Syntax | 6-4 | |  |
| Using a Subquery | 6-5 | |  |

**vii**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

|  |  |  |  |
| --- | --- | --- | --- |
| 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 6-19

Practice 6: Overview 6-20

1. **Using the Set Operators** Objectives 7-2

Set Operators 7-3

Tables Used in This Lesson 7-4 UNION Operator 7-8

Using the UNION Operator 7-9 UNION ALL Operator 7-11

Using the UNION ALL Operator 7-12 INTERSECT Operator 7-13

Using the INTERSECT Operator 7-14 MINUS Operator 7-15

Set Operator Guidelines 7-17

The Oracle Server and Set Operators 7-18 Matching the SELECT Statements 7-19 Matching the SELECT Statement: Example 7-20 Controlling the Order of Rows 7-21 Summary 7-22

Practice 7: Overview 7-23

1. **Manipulating Data** Objectives 8-2

Data Manipulation Language 8-3 Adding a New Row to a Table 8-4 INSERT Statement Syntax 8-5 Inserting New Rows 8-6

**viii**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

Inserting Rows with Null Values 8-7

Inserting Special Values 8-8

Inserting Specific Date Values 8-9

Creating a Script 8-10

Copying Rows from Another Table 8-11

Changing Data in a Table 8-12

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

Removing a Row from a Table 8-17

DELETE Statement 8-18

Deleting Rows from a Table 8-19

Deleting Rows Based on Another Table 8-20

TRUNCATE Statement 8-21

Using a Subquery in an INSERT Statement 8-22

Database Transactions 8-24

Advantages of COMMIT and ROLLBACK Statements 8-26 Controlling Transactions 8-27

Rolling Back Changes to a Marker 8-28

Implicit Transaction Processing 8-29

State of the Data Before COMMIT or ROLLBACK 8-31 State of the Data After COMMIT 8-32

Committing Data 8-33

State of the Data After ROLLBACK 8-34

Statement-Level Rollback 8-36

Read Consistency 8-37

Implementation of Read Consistency 8-38

Summary 8-39

Practice 8: Overview 8-40

1. **Using DDL Statements to Create and Manage Tables** Objectives 9-2

Database Objects 9-3 Naming Rules 9-4

CREATE TABLE Statement 9-5 Referencing Another User’s Tables 9-6 DEFAULT Option 9-7

Creating Tables 9-8 Data Types 9-9

**ix**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Datetime Data Types | | 9-11 | |  |  |
| Including Constraints | | 9-17 | |  |  |
| Constraint Guidelines | | 9-18 | |  |  |
| Defining Constraints 9-19 | | |  |  |  |
| NOT NULL Constraint | | 9-21 | |  |  |
| UNIQUE Constraint 9-22 | | |  |  |  |
| PRIMARY KEY Constraint | | | 9-24 |  |  |
| FOREIGN KEY Constraint | | | 9-25 |  |  |
| FOREIGN KEY Constraint: Keywords | | | | 9 -27 | |
| CHECK Constraint | 9-28 | |  |  |  |
| CREATE TABLE: Example 9-29 | | | |  |  |
| Violating Constraints | | 9-30 |  |  |  |
| Creating a Table by Using a Subquery | | | | 9-32 | |
| ALTER TABLE Statement | | | 9-34 |  |  |
| Dropping a Table | 9-35 | |  |  |  |
| Summary 9-36 |  |  |  |  |  |
| Practice 9: Overview | | 9-37 |  |  |  |
| **10 Creating Other Schema Objects** | | | |  |  |
| Objectives 10-2 |  |  |  |  |  |
| Database Objects | 10-3 | |  |  |  |
| What Is a View? | 10-4 | |  |  |  |
| Advantages of Views | | 10-5 | |  |  |
| Simple Views and Complex Views 10-6 | | | | |  |
| Creating a View 10-7 | |  |  |  |  |
| Retrieving Data from a View 10-10 | | | |  |  |
| Modifying a View | 10-11 | |  |  |  |
| Creating a Complex View | | | 10-12 |  |  |
| Rules for Performing DML Operations on a View 10-13 | | | | | |
| Using the WITH CHECK OPTION Clause | | | | | 10-16 |
| Denying DML Operations | | | 10-18 |  |  |
| Removing a View | 10-20 | |  |  |  |
| Practice 10: Overview of Part 1 10-21 | | | |  |  |
| Sequences 10-22 | |  |  |  |  |
| CREATE SEQUENCE Statement: Syntax | | | | | 10-24 |
| Creating a Sequence | | 10-25 | |  |  |
| NEXTVAL and CURRVAL Pseudocolumns | | | | | 10-26 |
| Using a Sequence 10-28 | | |  |  |  |
| Caching Sequence Values | | | 10-29 |  |  |
| Modifying a Sequence | | 10-30 | |  |  |

**x**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

Guidelines for Modifying a Sequence 10-31

Indexes 10-33

How Are Indexes Created? 10-35

Creating an Index 10-36

Index Creation Guidelines 10-37

Removing an Index 10-38

Synonyms 10-39

Creating and Removing Synonyms 10-41

Summary 10-42

Practice 10: Overview of Part 2 10-43

1. **Managing Objects with Data Dictionary Views** Objectives 11-2

The Data Dictionary 11-3 Data Dictionary Structure 11-4

How to Use the Dictionary Views 11-6

USER\_OBJECTS and ALL\_OBJECTS Views 11-7 USER\_OBJECTS View 11-8

Table Information 11-9 Column Information 11-10 Constraint Information 11-12 View Information 11-15 Sequence Information 11-16 Synonym Information 11-18 Adding Comments to a Table 11-19 Summary 11-20

Practice 11: Overview 11-21

1. **Practice Solutions**
2. **Table Descriptions and Data**
3. **Oracle Join Syntax**
4. **Using SQL\*Plus**
5. **Using SQL Developer**

**Index**

**xi**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices**

**Additional Practices: Table Descriptions and Data**

**Additional Practices: Solutions**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**xii**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Additional Practices**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

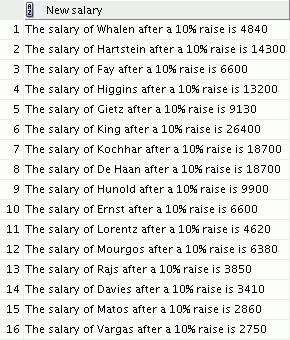
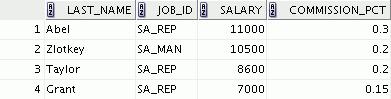
|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices**

These exercises can be used for extra practice after you have discussed the following topics:

basic SQL SELECT statement and SQL functions.

1. The HR department needs to find data for all the clerks who were hired after 1997.
2. The HR department needs a report of employees who earn commission. Show the last name, job, salary, and commission of these employees. Sort the data by salary in descending order.
3. For budgeting purposes, the HR department needs a report on projected raises. The report should display those employees who have no commission but who have a 10% raise in salary (round off the salaries).

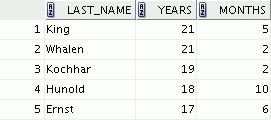


|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 2**

**Additional Practices (continued)**

1. Create a report of employees and their duration of employment. Show the last names of all employees together with the number of years and the number of completed months that they have been employed. Order the report by the duration of their employment. The employee who has been employed the longest should appear at the top of the list.



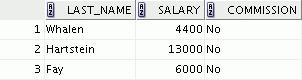
**…**

****

5. Show those employees who have a last name starting with the letters *J*, *K*, *L*, or *M*.



1. Create a report that displays all employees and indicate with the words *Yes* or *No* whether they receive a commission. Use the DECODE expression in your query.



**…**

****

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

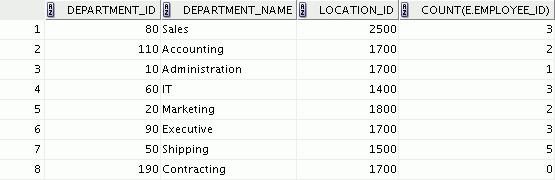
**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 3**

**Additional Practices (continued)**

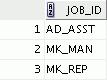
These exercises can be used for extra practice after you have discussed the following topics:

basic SQL SELECT statement, SQL functions, joins, and group functions.

1. Create a report that displays the department name, location, name, job title, and salary of those employees who work in a specific location. Prompt the user for the location. For example, if the user enters 1800, the following are the results:
2. Find the number of employees who have a last name that ends with the letter *n*. Create two possible solutions.
3. Create a report that shows the name, location, and number of employees for each department. Make sure that the report also includes departments without employees.



1. The HR department needs to find the job titles in departments 10 and 20. Create a report to display the job IDs for those departments.
2. Create a report that displays the jobs that are found in the Administration and Executive departments. Also display the number of employees for these jobs. Show the job with the highest number of employees first.



|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

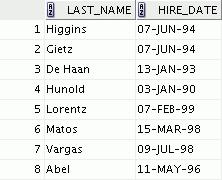
**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 4**

**Additional Practices (continued)**

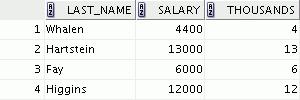
These exercises can be used for extra practice after you have discussed the following topics:

basic SQL SELECT statements, SQL functions, joins, group functions, and subqueries.

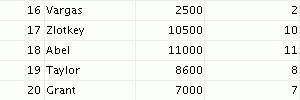
1. Show all employees who were hired in the first half of the month (before the 16th of the month).



1. Create a report that displays the following for all employees: last name, salary, and salary expressed in terms of thousands of dollars.



**…**

****

1. Show all employees who have managers with a salary higher than $15,000. Show the following data: employee name, manager name, manager salary, and salary grade of the manager.



|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

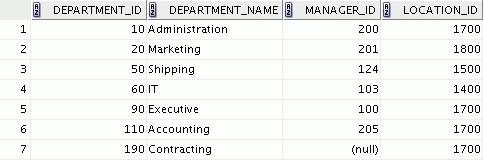
**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 5**

**Additional Practices (continued)**

1. Show the department number, name, number of employees, and average salary of all departments along with the names, salaries, and jobs of the employees working in each department.



1. Create a report to display the department number and the lowest salary of the department with the highest average salary.
2. Create a report that displays the departments where no sales representatives work. Include the department number, department name, and location in the output.

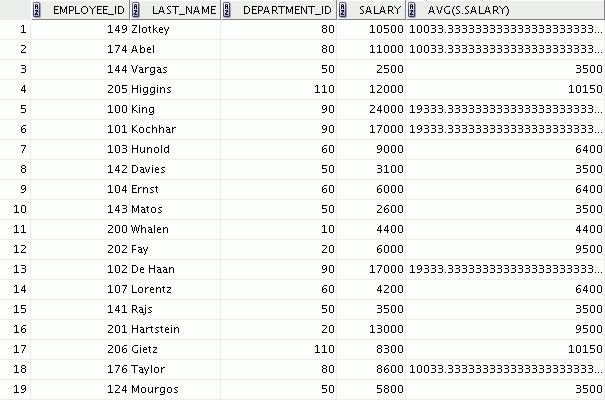
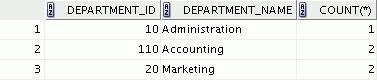


|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 6**

**Additional Practices (continued)**

1. Create the following statistical reports for the HR department: Include the department number, department name, and the number of employees working in each department that:
   1. Employs fewer than three employees:
   2. Has the highest number of employees:
   3. Has the lowest number of employees:
2. Create a report that displays the employee number, last name, salary, department number, and the average salary in their departments for all employees.

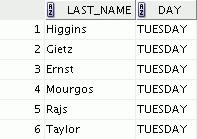


|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

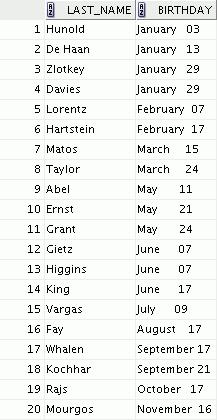
**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 7**

**Additional Practices (continued)**

1. Show all employees who were hired on the day of the week on which the highest number of employees were hired.



1. Create an anniversary overview based on the hire date of the employees. Sort the anniversaries in ascending order.



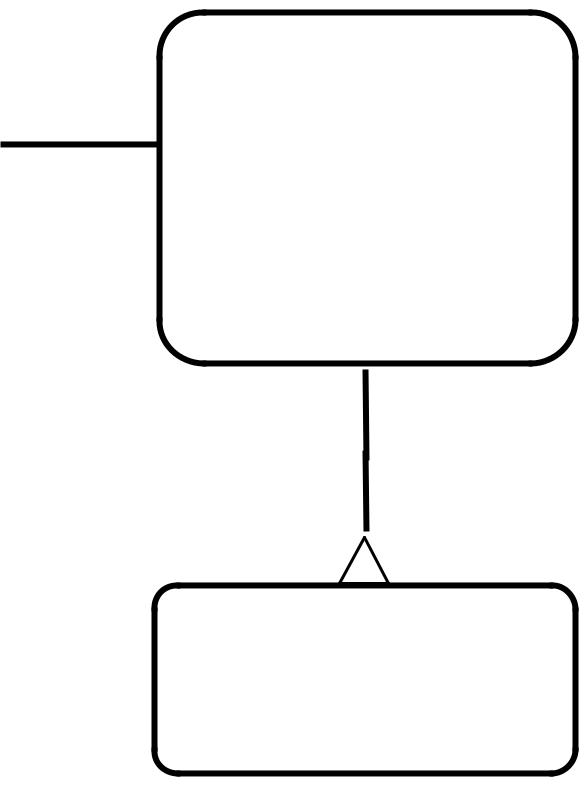
|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 8**

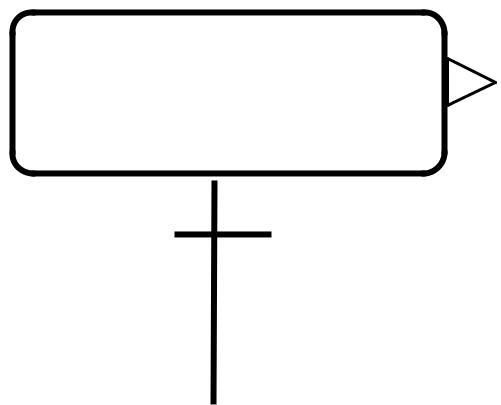
**Additional Practices: Case Study**

In this case study, you build a set of database tables for a video application. After you create the tables, you insert, update, and delete records in a video store database and generate a report. The database contains only the essential tables.

The following is a diagram of the entities and attributes for the video application:



RESERVATION #\* res date

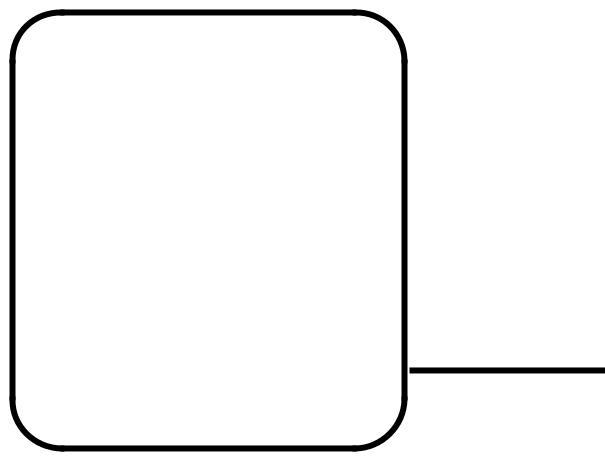


**set up for**

****

**responsible**

**for**

****

MEMBER

#\* id

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **for** | | | TITLE | | |
| #\* id | | |
|  |  | |
|  |  |  | \* title | | |
|  | **the subject** | | \* description | | |
|  |  | **of** | o rating | | |
|  |  |  | o category | | |
|  |  |  | o release date | | |
|  |  |  |  | **available as** | |
|  |  |  | **a copy** |  |  |
|  |  |  |  |
|  |  |  | TITLE\_COPY | | |
|  |  |  | #\* id | | |
|  |  |  | \* status | | |

|  |
| --- |
| Oracle Academy Use Only |

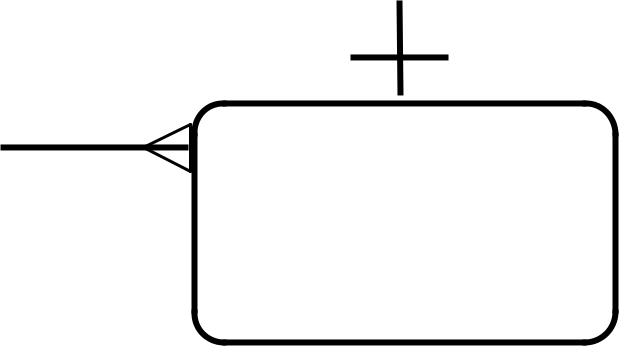
* + last name o first name o address o city

o phone

* join date

**responsible for**

 **the subject of**

****

**made against**

|  |  |
| --- | --- |
|  | RENTAL |
| **created** | #\* book date |
| **for** | o act ret date |
|  | o exp ret date |

|  |
| --- |
| Oracle Internal & |

**Note:** If you want to build the tables, you can execute the commands in thebuildtab.sql script in SQL Developer. If you want to drop the tables, you can execute the commands in the dropvid.sql script in SQL Developer. Then you can execute the commands in the buildvid.sql script in SQL Developer to create and populate the tables.

* If you use the buildtab.sql script to build the tables, start with step 4.
* If you use the dropvid.sql script to remove the video tables, start with step 1.
* If you use the buildvid.sql script to build and populate the tables, start with step 6(b).

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 9**

**Additional Practices: Case Study (continued)**

1. Create the tables based on the following table instance charts. Choose the appropriate data types and ensure that you add integrity constraints.

a. Table name: MEMBER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Column\_** | MEMBER\_ | LAST\_ | FIRST\_NAME | ADDRESS | CITY | PHONE | JOIN |
| **Name** | ID | NAME |  |  |  |  | \_ |
|  |  |  |  |  |  |  | DATE |
| **Key** | PK |  |  |  |  |  |  |
| **Type** |  |  |  |  |  |  |  |
| **Null/** | NN,U | NN |  |  |  |  | NN |
| **Unique** |  |  |  |  |  |  |  |
| **Default** |  |  |  |  |  |  | System |
| **Value** |  |  |  |  |  |  | Date |
|  |  |  |  |  |  |  |  |
| **Data** | NUMBER | VARCHAR2 | VARCHAR2 | VARCHAR2 | VARCHAR2 | VARCHAR2 | DATE |
| **Type** |  |  |  |  |  |  |  |
| **Length** | 10 | 25 | 25 | 100 | 30 | 15 |  |
|  |  |  |  |  |  |  |  |

b. Table name: TITLE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column\_** | TITLE\_ID | TITLE | DESCRIPTION | RATING | CATEGORY | RELEASE\_ |
| **Name** |  |  |  |  |  | DATE |
|  |  |  |  |  |  |  |
| **Key** | PK |  |  |  |  |  |
| **Type** |  |  |  |  |  |  |
| **Null/** | NN,U | NN | NN |  |  |  |
| **Unique** |  |  |  |  |  |  |
| **Check** |  |  |  | G, PG, R, | DRAMA, |  |
|  |  |  |  | NC17, NR | COMEDY, |  |
|  |  |  |  |  | ACTION, |  |
|  |  |  |  |  | CHILD, |  |
|  |  |  |  |  | SCIFI, |  |
|  |  |  |  |  | DOCUMEN |  |
|  |  |  |  |  | TARY |  |
| **Data Type** | NUMBER | VARCHAR2 | VARCHAR2 | VARCHAR2 | VARCHAR2 | DATE |
|  |  |  |  |  |  |  |
| **Length** | 10 | 60 | 400 | 4 | 20 |  |
|  |  |  |  |  |  |  |

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 10**

**Additional Practices: Case Study (continued)**

c. Table name: TITLE\_COPY

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | COPY\_ID | TITLE\_ID | STATUS |
| **Name** |  |  |  |
| **Key** | PK | PK,FK |  |
| **Type** |  |  |  |
| **Null/** | NN,U | NN,U | NN |
| **Unique** |  |  |  |
| **Check** |  |  | AVAILABLE, |
|  |  |  | DESTROYED, |
|  |  |  | RENTED, |
|  |  |  | RESERVED |
| **FK Ref** |  | TITLE |  |
| **Table** |  |  |  |
| **FK Ref** |  | TITLE\_ID |  |
| **Col** |  |  |  |
| **Data** | NUMBER | NUMBER | VARCHAR2 |
| **Type** |  |  |  |
| **Length** | 10 | 10 | 15 |
|  |  |  |  |

d. Table name: RENTAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | BOOK\_ | MEMBER\_ | COPY\_ | ACT\_RET\_ | EXP\_RET\_ | TITLE\_ |
| **Name** | DATE | ID | ID | DATE | DATE | ID |
| **Key** | PK | PK,FK1 | PK,FK2 |  |  | PK,FK2 |
| **Type** |  |  |  |  |  |  |
| **Default** | System |  |  |  | System Date |  |
| **Value** | Date |  |  |  | + 2 days |  |
| **FK Ref** |  | MEMBER | TITLE\_ |  |  | TITLE\_ |
| **Table** |  |  | COPY |  |  | COPY |
| **FK Ref** |  | MEMBER\_I | COPY\_ |  |  | TITLE\_ID |
| **Col** |  | D | ID |  |  |  |
| **Data** | DATE | NUMBER | NUMBER | DATE | DATE | NUMBER |
| **Type** |  |  |  |  |  |  |
| **Length** |  | 10 | 10 |  |  | 10 |
|  |  |  |  |  |  |  |

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 11**

**Additional Practices: Case Study (continued)**

1. Table name: RESERVATION

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | RES\_ | MEMBER\_ | TITLE\_ |
| **Name** | DATE | ID | ID |
| **Key** | PK | PK,FK1 | PK,FK2 |
| **Type** |  |  |  |
| **Null/** | NN,U | NN,U | NN |
| **Unique** |  |  |  |
| **FK Ref** |  | MEMBER | TITLE |
| **Table** |  |  |  |
|  |  |  |  |
| **FK Ref** |  | MEMBER\_ID | TITLE\_ID |
| **Column** |  |  |  |
| **Data Type** | DATE | NUMBER | NUMBER |
|  |  |  |  |
| **Length** |  | 10 | 10 |

2. Verify that the tables and constraints were created properly by checking the data dictionary.



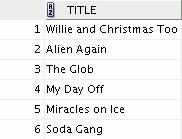
|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 12**

**Additional Practices: Case Study (continued)**

1. Create sequences to uniquely identify each row in the MEMBER table and the TITLE table.
   1. Member number for the MEMBER table: Start with 101; do not allow caching of values. Name the sequence MEMBER\_ID\_SEQ.
   2. Title number for the TITLE table: Start with 92; do not allow caching of values. Name the sequence TITLE\_ID\_SEQ.
   3. Verify the existence of the sequences in the data dictionary.
2. Add data to the tables. Create a script for each set of data to be added.
   1. Add movie titles to the TITLE table. Write a script to enter the movie information. Save the statements in a script named lab\_apcs\_4a.sql. Use the sequences to uniquely identify each title. Enter the release dates in the DD-MON-YYYY format.

Remember that single quotation marks in a character field must be specially handled. Verify your additions.



|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 13**

**Additional Practices: Case Study (continued)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Description** | **Rating** | **Category** | **Release\_date** |
| Willie and | All of Willie’s friends make a | G | CHILD | 05-OCT-1995 |
| Christmas Too | Christmas list for Santa, but |  |  |  |
|  | Willie is yet to add his own |  |  |  |
|  | wish list. |  |  |  |
| Alien Again | Yet another installation of | R | SCIFI | 19-MAY-1995 |
|  | science fiction history. Can |  |  |  |
|  | the heroine save the planet |  |  |  |
|  | from the alien life form? |  |  |  |
| The Glob | A meteor crashes near a small | NR | SCIFI | 12-AUG-1995 |
|  | American town and unleashes |  |  |  |
|  | carnivorous goo in this classic. |  |  |  |
| My Day Off | With a little luck and a lot of | PG | COMEDY | 12-JUL-1995 |
|  | ingenuity, a teenager skips |  |  |  |
|  | school for a day in New York. |  |  |  |
| Miracles on Ice | A six-year-old has doubts | PG | DRAMA | 12-SEP-1995 |
|  | about Santa Claus, but she |  |  |  |
|  | discovers that miracles really |  |  |  |
|  | do exist. |  |  |  |
| Soda Gang | After discovering a cache of | NR | ACTION | 01-JUN-1995 |
|  | drugs, a young couple find |  |  |  |
|  | themselves pitted against a |  |  |  |
|  | vicious gang. |  |  |  |

1. Add data to the MEMBER table. Place the INSERT statements in a script named

lab\_apcs\_4b.sql. Execute the commands in the script. Be sure to use the sequence to add the member numbers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **First\_** |  |  |  |  |  |
| **Name** | **Last\_Name** | **Address** | **City** | **Phone** | **Join\_Date** |
| Carmen | Velasquez | 283 King | Seattle | 206-899-6666 | 08-MAR-1990 |
|  |  | Street |  |  |  |
|  |  |  |  |  |  |
| LaDoris | Ngao | 5 Modrany | Bratislava | 586-355-8882 | 08-MAR-1990 |
|  |  |  |  |  |  |
| Midori | Nagayama | 68 Via | Sao Paolo | 254-852-5764 | 17-JUN-1991 |
|  |  | Centrale |  |  |  |
|  |  |  |  |  |  |
| Mark | Quick-to-See | 6921 King | Lagos | 63-559-7777 | 07-APR-1990 |
|  |  | Way |  |  |  |
|  |  |  |  |  |  |
| Audry | Ropeburn | 86 Chu Street | Hong Kong | 41-559-87 | 18-JAN-1991 |
|  |  |  |  |  |  |
| Molly | Urguhart | 3035 Laurier | Quebec | 418-542-9988 | 18-JAN-1991 |
|  |  |  |  |  |  |

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 14**

**Additional Practices: Case Study (continued)**

1. Add the following movie copies in the TITLE\_COPY table: **Note:** Have theTITLE\_IDnumbers available for this exercise.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Copy\_Id** | **Status** | **Title** | **Copy\_Id** |
|  |  |  |  |  |
| Willie and | 1 | AVAILABLE | Willie and Christmas | 1 |
| Christmas Too |  |  | Too |  |
| Alien Again | 1 | AVAILABLE | Alien Again | 1 |
|  |  |  |  |  |
|  | 2 | RENTED |  | 2 |
|  |  |  |  |  |
| The Glob | 1 | AVAILABLE | The Glob | 1 |
|  |  |  |  |  |
| My Day Off | 1 | AVAILABLE | My Day Off | 1 |
|  |  |  |  |  |
|  | 2 | AVAILABLE |  | 2 |
|  |  |  |  |  |
|  | 3 | RENTED |  | 3 |
|  |  |  |  |  |
| Miracles on Ice | 1 | AVAILABLE | Miracles on Ice | 1 |
|  |  |  |  |  |
| Soda Gang | 1 | AVAILABLE | Soda Gang | 1 |
|  |  |  |  |  |

1. Add the following rentals to the RENTAL table:

**Note:** The title number may be different depending on the sequence number.

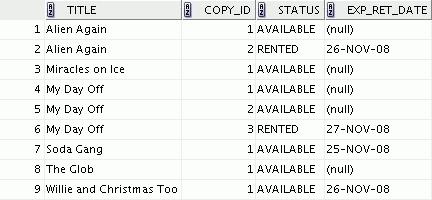
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title\_ Id** | **Copy\_** | **Member\_Id** |  |  |
|  | **Id** |  | **Book\_date** | **Exp\_Ret\_Date** |
| 92 | 1 | 101 | 3 days ago | 1 day ago |
|  |  |  |  |  |
| 93 | 2 | 101 | 1 day ago | 1 day from now |
|  |  |  |  |  |
| 95 | 3 | 102 | 2 days ago | Today |
|  |  |  |  |  |
| 97 | 1 | 106 | 4 days ago | 2 days ago |
|  |  |  |  |  |

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

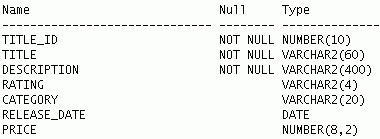
**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 15**

**Additional Practices: Case Study (continued)**

1. Create a view named TITLE\_AVAIL to show the movie titles, the availability of each copy, and its expected return date if rented. Query all rows from the view. Order the results by title. **Note:** Your results may be different.



1. Make changes to the data in the tables.
   1. Add a new title. The movie is “Interstellar Wars,” which is rated PG and classified as a science fiction movie. The release date is 07-JUL-77. The description is “Futuristic interstellar action movie. Can the rebels save the humans from the evil empire?” Be sure to add a title copy record for two copies.
   2. Enter two reservations. One reservation is for Carmen Velasquez, who wants to rent “Interstellar Wars.” The other is for Mark Quick-to-See, who wants to rent “Soda Gang.”
2. Make a modification to one of the tables.
   1. Run the script in lab\_apcs\_7a.sql to add a PRICE column to the TITLE table to record the purchase price of the video. Verify your modifications.



|  |  |
| --- | --- |
| **Title** | **Price** |
|  |  |
| Willie and Christmas Too | 25 |
|  |  |
| Alien Again | 35 |
|  |  |
| The Glob | 35 |
|  |  |
| My Day Off | 35 |
|  |  |
| Miracles on Ice | 30 |
|  |  |
| Soda Gang | 35 |
|  |  |
| Interstellar Wars | 29 |
|  |  |

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

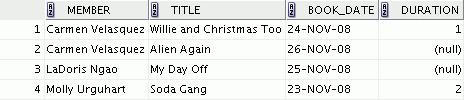
**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 16**

* 1. Create a script named lab\_apcs\_7b.sql that contains UPDATE statements that update each video with a price according to the preceding list. Run the commands in the script.

**Note:** Have theTITLE\_IDnumbers available for this exercise.

1. Create a report that contains each customer’s history of renting videos. Be sure to include the customer name, movie rented, dates of the rental, and duration of rentals. Total the number of rentals for all customers for the reporting period. Save the commands that generate the report in a script file named lab\_apcs\_8.sql.

**Note:** Your results may be different.



|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices - 17**

Oracle Internal & Oracle Academy Use Only



**Additional Practices**

****

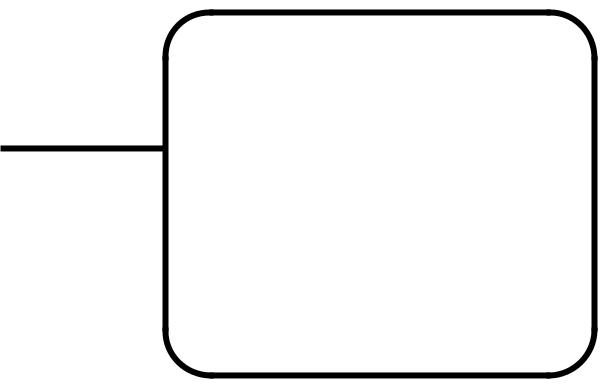
**Table Descriptions and Data**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

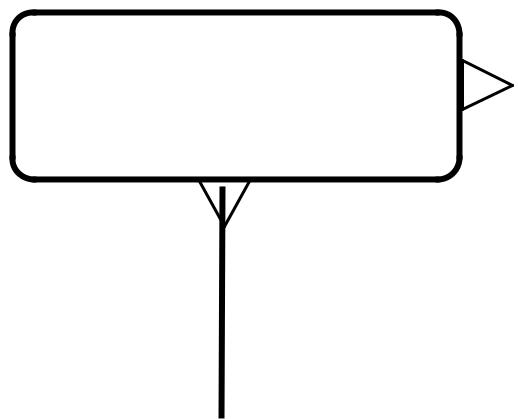
**Tables Used in Additional Practices**

Additional practice questions 1-21 use the HR schema. Refer to Appendix B to look at the HR schema tables. The tables used in the additional practices: case study are described below.

Note: These table do not exist by default. You will be creating them in the case study practice questions.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **for** | | | TITLE |
|  |  |  | #\* id |
| RESERVATION | | |  |  |  |
|  |  |  | \* title |
|  |  |  |
| #\* res date | | |  | **the subject** | | \* description |
|  |
|  |  |  |  |  | **of** | o rating |
|  | **set up for** | |  |  |  | o category |
|  |  |  |  |  |  | o release date |
|  |  |  |  |  |  |

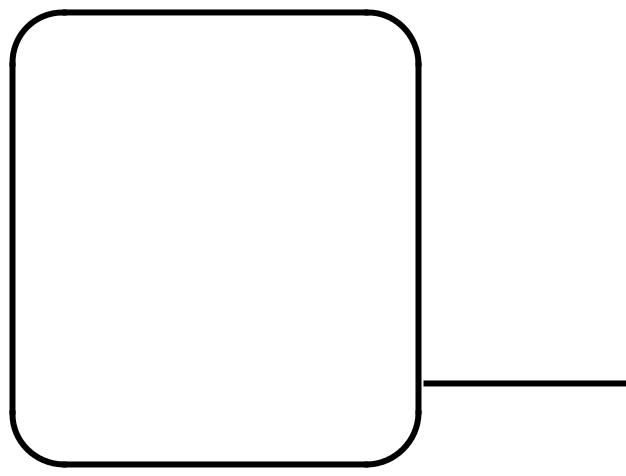


|  |
| --- |
| Only |

**responsible**

****

**for**

****

MEMBER

#\* id

* last name o first name

|  |  |
| --- | --- |
| o address | **responsible** |
| o city | **for** |
| o phone |  |
| \* join date |  |

 **available as**

****

**a copy**

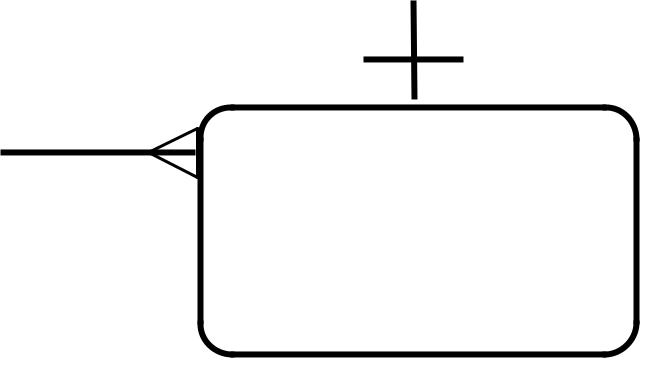
****

TITLE\_COPY

#\* id

\* status

 **the subject of**

****

**made against**

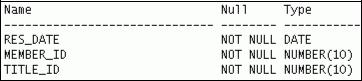
|  |  |
| --- | --- |
|  | RENTAL |
| **created** | #\* book date |
| **for** | o act ret date |
|  | o exp ret date |

|  |
| --- |
| Oracle Internal & Oracle Academy Use |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices Tables - 2**

**RESERVATION Table**

DESCRIBE reservation

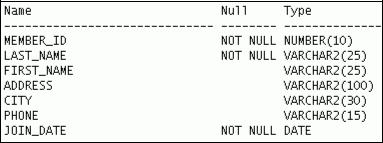


|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices Tables - 3**

**MEMBER Table**

DESCRIBE member

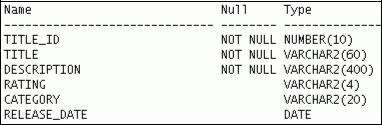


|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices Tables - 4**

**TITLE Table**

DESCRIBE title

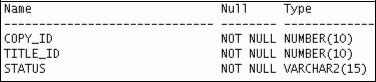


|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices Tables - 5**

**TITLE\_COPY Table**

DESCRIBE title\_copy

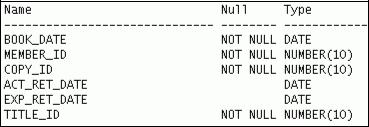


|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices Tables - 6**

**RENTAL Table**

DESCRIBE rental



|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices Tables - 7**

Oracle Internal & Oracle Academy Use Only

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Additional Practices:**

**Solutions**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices: Solutions**

These exercises can be used for extra practice after you have discussed the following topics:

basic SQL SELECT statement and SQL functions.

1. The HR department needs to find data for all the clerks who were hired after 1997.

SELECT \*

FROM employees

WHERE job\_id = 'ST\_CLERK'

AND hire\_date > '31-DEC-1997';

1. The HR department needs a report of employees who earn commission. Show the last name, job, salary, and commission of these employees. Sort the data by salary in descending order.

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

FROM employees

WHERE commission\_pct IS NOT NULL

ORDER BY salary DESC;

1. For budgeting purposes, the HR department needs a report on projected raises. The report should display those employees who have no commission but who have a 10% raise in salary (round off the salaries).

SELECT 'The salary of '||last\_name||' after a 10% raise is '

* + ROUND(salary\*1.10) "New salary" FROM employees

WHERE commission\_pct IS NULL;

1. Create a report of employees and their duration of employment. Show the last names of all the employees along with the number of years and the number of completed months that they have been employed. Order the report by the duration of their employment. The employee who has been employed the longest should appear at the top of the list.

|  |  |
| --- | --- |
| SELECT last\_name, |  |
| TRUNC(MONTHS\_BETWEEN(SYSDATE, hire\_date) / 12) YEARS, |  |
| TRUNC(MOD(MONTHS\_BETWEEN(SYSDATE, hire\_date), 12)) | MONTHS |
| FROM employees |  |
| ORDER BY years DESC, MONTHS desc; |  |
|  |  |

5. Show those employees who have a last name starting with the letters *J*, *K*, *L*, or *M*.

SELECT last\_name

FROM employees

WHERE SUBSTR(last\_name, 1,1) IN ('J', 'K', 'L', 'M');

1. Create a report that displays all the employees and indicate with the words *Yes* or *No* whether they receive a commission. Use the DECODE expression in your query.

SELECT last\_name, salary,

decode(commission\_pct, NULL, 'No', 'Yes') commission FROM employees;

**Oracle Database 10*g*: SQL Fundamentals I** **Additional Practices: Solutions - 2**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices: Solutions (continued)**

These exercises can be used for extra practice after you have discussed the following topics:

basic SQL SELECT statement, SQL functions, joins, and group functions.

1. Create a report that displays the department name, location, name, job title, and salary of those employees who work in a specific location. Prompt the user for the location.

|  |  |  |  |
| --- | --- | --- | --- |
| SELECT | d.department\_name, | | d.location\_id, e.last\_name, e.job\_id, e.salary |
| FROM | employees e, departments d | | |
| WHERE | e.department\_id | = | d.department\_id |
| AND | d.location\_id = | &dept\_no; | |
|  |  |  |  |

1. Find the number of employees who have a last name that ends with the letter *n*. Create two possible solutions.

SELECT COUNT(\*)

FROM employees

WHERE last\_name LIKE '%n';

--or

SELECT COUNT(\*)

FROM employees

WHERE SUBSTR(last\_name, -1) = 'n';

1. Create a report that shows the name, location, and number of employees for each department. Make sure that the report also includes departments without employees.

|  |  |  |
| --- | --- | --- |
| SELECT | d.department\_id, | d.department\_name, |
|  | d.location\_id, | COUNT(e.employee\_id) |
| FROM | employees e RIGHT | OUTER JOIN departments d |
| ON | e.department\_id = | d.department\_id |

GROUP BY d.department\_id, d.department\_name, d.location\_id;

1. The HR department needs to find the job titles in departments 10 and 20. Create a report to display the job IDs for these departments.

SELECT DISTINCT job\_id

FROM employees

WHERE department\_id IN (10, 20);

1. Create a report that displays the jobs that are found in the Administration and Executive departments. Also display the number of employees for these jobs. Show the job with the highest number of employees first.

SELECT e.job\_id, count(e.job\_id) FREQUENCY

FROM employees e JOIN departments d

ON e.department\_id = d.department\_id

WHERE d.department\_name IN ('Administration', 'Executive')

GROUP BY e.job\_id

ORDER BY FREQUENCY DESC;

These exercises can be used for extra practice after you have discussed the following topics:

basic SQL SELECT statements, SQL functions, joins, group functions, and subqueries.

1. Show all employees who were hired in the first half of the month (before the 16th of the month).

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I** **Additional Practices: Solutions - 3**

**Additional Practices: Solutions (continued)**

SELECT last\_name, hire\_date

FROM employees

WHERE TO\_CHAR(hire\_date, 'DD') < 16;

1. Create a report that displays the following for all employees: last name, salary, and salary expressed in terms of thousands of dollars.

SELECT last\_name, salary, TRUNC(salary, -3)/1000 Thousands FROM employees;

1. Show all employees who have managers with a salary higher than $15,000. Show the following data: employee name, manager name, manager salary, and salary grade of the manager.

|  |  |  |
| --- | --- | --- |
| SELECT | e.last\_name, | m.last\_name manager, m.salary, j.grade\_level |
| FROM | employees e JOIN employees m | |
| ON | e.manager\_id | = m.employee\_id |
| JOIN | job\_grades j |  |
| ON | m.salary BETWEEN j.lowest\_sal AND j.highest\_sal | |
| AND | m.salary > 15000; | |
|  |  |  |

1. Show the department number, name, number of employees, and average salary of all departments together with the names, salaries, and jobs of the employees working in each department.

|  |  |  |
| --- | --- | --- |
| SELECT | d.department\_id, d.department\_name, | |
|  | count(e1.employee\_id) employees, | |
|  | NVL(TO\_CHAR(AVG(e1.salary), '99999.99'), 'No average' ) avg\_sal, | |
|  | e2.last\_name, | e2.salary, e2.job\_id |
| FROM | departments d | RIGHT OUTER JOIN employees e1 |
| ON | d.department\_id = e1.department\_id | |
| RIGHT | OUTER JOIN employees e2 | |
| ON | d.department\_id | = e2.department\_id |

GROUP BY d.department\_id, d.department\_name, e2.last\_name, e2.salary, e2.job\_id

ORDER BY d.department\_id, employees;

1. Create a report to display the department number and lowest salary of the department with the highest average salary.

SELECT department\_id, MIN(salary)

FROM employees

GROUP BY department\_id

HAVING AVG(salary) = (SELECT MAX(AVG(salary))

FROM employees

GROUP BY department\_id);

1. Create a report that displays the departments where no sales representatives work. Include the department number, department name, and location in the output.

SELECT \*

FROM departments

WHERE department\_id NOT IN(SELECT department\_id

FROM employees

**Oracle Database 10*g*: SQL Fundamentals I** **Additional Practices: Solutions - 4**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices: Solutions (continued)**

WHERE job\_id = 'SA\_REP'

AND department\_id IS NOT NULL);

1. Create the following statistical reports for the HR department: Include the department number, department name, and the number of employees working in each department that:

a. Employs fewer than three employees:

|  |  |  |
| --- | --- | --- |
| SELECT | d.department\_id, d.department\_name, COUNT(\*) | |
| FROM | departments d JOIN | employees e |
| ON | d.department\_id = e.department\_id | |
| GROUP BY d.department\_id, | | d.department\_name |
| HAVING | COUNT(\*) < 3; |  |
|  |  |  |

b. Has the highest number of employees:

|  |  |  |
| --- | --- | --- |
| SELECT | d.department\_id, d.department\_name, COUNT(\*) | |
| FROM | departments d JOIN | employees e |
| ON | d.department\_id = e.department\_id | |
| GROUP BY d.department\_id, | | d.department\_name |
| HAVING | COUNT(\*) = (SELECT | MAX(COUNT(\*)) |
|  | FROM | employees |
|  | GROUP BY department\_id); | |
|  |  |  |

c. Has the lowest number of employees:

|  |  |  |
| --- | --- | --- |
| SELECT | d.department\_id, d.department\_name, COUNT(\*) | |
| FROM | departments d JOIN | employees e |
| ON | d.department\_id = e.department\_id | |
| GROUP BY d.department\_id, | | d.department\_name |
| HAVING | COUNT(\*) = (SELECT | MIN(COUNT(\*)) |
|  | FROM | employees |
|  | GROUP BY department\_id); | |
|  |  |  |

1. Create a report that displays the employee number, last name, salary, department number, and the average salary in their department for all employees.

|  |  |  |
| --- | --- | --- |
| SELECT | e.employee\_id, e.last\_name, e.department\_id, e.salary, | |
| AVG(s.salary) | |  |
| FROM | employees e JOIN | employees s |
| ON | e.department\_id = s.department\_id | |
| GROUP BY e.employee\_id, | | e.last\_name, e.department\_id, e.salary; |
|  |  |  |

1. Show all employees who were hired on the day of the week on which the highest number of employees were hired.

SELECT last\_name, TO\_CHAR(hire\_date, 'DAY') day

FROM employees

WHERE TO\_CHAR(hire\_date, 'Day') =

(SELECT TO\_CHAR(hire\_date, 'Day')

FROM employees

GROUP BY TO\_CHAR(hire\_date, 'Day')

HAVING COUNT(\*) = (SELECT MAX(COUNT(\*))

FROM employees

GROUP BY TO\_CHAR(hire\_date, 'Day')));

**Oracle Database 10*g*: SQL Fundamentals I** **Additional Practices: Solutions - 5**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices: Solutions (continued)**

1. Create an anniversary overview based on the hire date of the employees. Sort the anniversaries in ascending order.

SELECT last\_name, TO\_CHAR(hire\_date, 'Month DD') BIRTHDAY

FROM employees

ORDER BY TO\_CHAR(hire\_date, 'DDD');

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I** **Additional Practices: Solutions - 6**

**Additional Practices: Case Study Solutions**

1. Create tables based on the following table instance charts. Choose the appropriate data types and be sure to add integrity constraints.

a. Table name: MEMBER

CREATE TABLE member

(member\_id NUMBER(10)

CONSTRAINT member\_member\_id\_pk PRIMARY KEY,

last\_name VARCHAR2(25)

CONSTRAINT member\_last\_ name\_nn NOT NULL,

first\_name VARCHAR2(25),

address VARCHAR2(100),

city VARCHAR2(30),

phone VARCHAR2(15),

join\_date DATE DEFAULT SYSDATE

CONSTRAINT member\_join\_date\_nn NOT NULL);

b. Table name: TITLE

|  |  |
| --- | --- |
| CREATE TABLE title |  |
| (title\_id | NUMBER(10) |
| CONSTRAINT | title\_title\_id\_pk PRIMARY KEY, |
| title | VARCHAR2(60) |
| CONSTRAINT | title\_title\_nn NOT NULL, |
| description | VARCHAR2(400) |
| CONSTRAINT | title\_description\_nn NOT NULL, |
| rating | VARCHAR2(4) |
| CONSTRAINT | title\_rating\_ck CHECK |
| (rating IN | ('G', 'PG', 'R', 'NC17', 'NR')), |
| category | VARCHAR2(20) |
| CONSTRAINT | title\_category\_ck CHECK |
| (category IN ('DRAMA', 'COMEDY', 'ACTION', | |
| 'CHILD', 'SCIFI', 'DOCUMENTARY')), | |
| release\_date | DATE); |
|  |  |

c. Table name: TITLE\_COPY

CREATE TABLE title\_copy

(copy\_id NUMBER(10),

title\_id NUMBER(10)

CONSTRAINT title\_copy\_title\_if\_fk REFERENCES title(title\_id),

status VARCHAR2(15)

CONSTRAINT title\_copy\_status\_nn NOT NULL CONSTRAINT title\_copy\_status\_ck CHECK (status IN ('AVAILABLE', 'DESTROYED','RENTED', 'RESERVED')),

CONSTRAINT title\_copy\_copy\_id\_title\_id\_pk PRIMARY KEY (copy\_id, title\_id));

d. Table name: RENTAL

**Oracle Database 10*g*: SQL Fundamentals I** **Additional Practices: Solutions - 7**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices: Case Study Solutions (continued)**

CREATE TABLE rental

(book\_date DATE DEFAULT SYSDATE,

member\_id NUMBER(10)

CONSTRAINT rental\_member\_id\_fk REFERENCES member(member\_id),

copy\_id NUMBER(10),

act\_ret\_date DATE,

exp\_ret\_date DATE DEFAULT SYSDATE + 2,

title\_id NUMBER(10),

CONSTRAINT rental\_book\_date\_copy\_title\_pk

PRIMARY KEY (book\_date, member\_id, copy\_id,title\_id),

CONSTRAINT rental\_copy\_id\_title\_id\_fk

FOREIGN KEY (copy\_id, title\_id)

REFERENCES title\_copy(copy\_id, title\_id));

1. Table name: RESERVATION

CREATE TABLE reservation

(res\_date DATE,

member\_id NUMBER(10)

CONSTRAINT reservation\_member\_id REFERENCES member(member\_id),

title\_id NUMBER(10)

CONSTRAINT reservation\_title\_id REFERENCES title(title\_id), CONSTRAINT reservation\_resdate\_mem\_tit\_pk PRIMARY KEY

(res\_date, member\_id, title\_id));

2. Verify that the tables and constraints were created properly by checking the data dictionary.

|  |  |  |  |
| --- | --- | --- | --- |
| SELECT | table\_name |  |  |
| FROM | user\_tables |  |  |
| WHERE | table\_name IN ('MEMBER', | 'TITLE', | 'TITLE\_COPY', |
|  | 'RENTAL', | 'RESERVATION'); | |
| SELECT | constraint\_name, constraint\_type, | | table\_name |
| FROM | user\_constraints |  |  |
| WHERE | table\_name IN ('MEMBER', | 'TITLE', | 'TITLE\_COPY', |
|  | 'RENTAL', | 'RESERVATION'); | |
|  |  |  |  |

3. Create sequences to uniquely identify each row in the MEMBER table and the TITLE table.

1. Member number for the MEMBER table: Start with 101; do not allow caching of values. Name the sequence MEMBER\_ID\_SEQ.

CREATE SEQUENCE member\_id\_seq

START WITH 101

NOCACHE;

1. Title number for the TITLE table: Start with 92; do not allow caching of values. Name the sequence TITLE\_ID\_SEQ.

CREATE SEQUENCE title\_id\_seq

START WITH 92

NOCACHE;

c. Verify the existence of the sequences in the data dictionary.

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I** **Additional Practices: Solutions - 8**

**Additional Practices: Case Study Solutions (continued)**

SELECT sequence\_name, increment\_by, last\_number

FROM user\_sequences

WHERE sequence\_name IN ('MEMBER\_ID\_SEQ', 'TITLE\_ID\_SEQ');

1. Add data to the tables. Create a script for each set of data to be added.
   1. Add movie titles to the TITLE table. Write a script to enter the movie information. Save the statements in a script named lab\_apcs\_4a.sql. Use the sequences to uniquely identify each title. Enter the release dates in the DD-MON-YYYY format.

Remember that single quotation marks in a character field must be specially handled. Verify your additions.

INSERT INTO title(title\_id, title, description, rating,

category, release\_date)

VALUES (title\_id\_seq.NEXTVAL, 'Willie and Christmas Too',

'All of Willie''s friends make a Christmas list for

Santa, but Willie has yet to add his own wish list.',

'G', 'CHILD', TO\_DATE('05-OCT-1995','DD-MON-YYYY'))

/

INSERT INTO title(title\_id , title, description, rating,

category, release\_date)

VALUES (title\_id\_seq.NEXTVAL, 'Alien Again', 'Yet another

installment of science fiction history. Can the

heroine save the planet from the alien life form?',

'R', 'SCIFI', TO\_DATE( '19-MAY-1995','DD-MON-YYYY'))

/

INSERT INTO title(title\_id, title, description, rating,

category, release\_date)

VALUES (title\_id\_seq.NEXTVAL, 'The Glob', 'A meteor crashes

near a small American town and unleashes carnivorous

goo in this classic.', 'NR', 'SCIFI',

TO\_DATE( '12-AUG-1995','DD-MON-YYYY'))

/

INSERT INTO title(title\_id, title, description, rating,

category, release\_date)

VALUES (title\_id\_seq.NEXTVAL, 'My Day Off', 'With a little

luck and a lot ingenuity, a teenager skips school for a day in New York.', 'PG', 'COMEDY', TO\_DATE( '12-JUL-1995','DD-MON-YYYY'))

/

...

COMMIT

/

SELECT title

FROM title;

1. Add data to the MEMBER table. Place the INSERT statements in a script named

lab\_apcs\_4b.sql. Execute the commands in the script. Be sure to use the sequence to add the member numbers.

SET VERIFY OFF

INSERT INTO member(member\_id, first\_name, last\_name, address, city, phone, join\_date)

**Oracle Database 10*g*: SQL Fundamentals I** **Additional Practices: Solutions - 9**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices: Case Study Solutions (continued)**

VALUES (member\_id\_seq.NEXTVAL, 'Carmen', 'Velasquez',

'283 King Street', 'Seattle', '206-899-6666', TO\_DATE('08-MAR-

1990',

'DD-MM-YYYY'))

/

INSERT INTO member(member\_id, first\_name, last\_name, address, city, phone, join\_date)

VALUES (member\_id\_seq.NEXTVAL, 'LaDoris', 'Ngao',

'5 Modrany', 'Bratislava', '586-355-8882', TO\_DATE('08-MAR-1990', 'DD-MM-YYYY'))

/

INSERT INTO member(member\_id, first\_name, last\_name, address, city, phone, join\_date)

VALUES (member\_id\_seq.NEXTVAL, 'Midori', 'Nagayama',

'68 Via Centrale', 'Sao Paolo', '254-852-5764', TO\_DATE('17-JUN-

1991',

'DD-MM-YYYY'))

/

INSERT INTO member(member\_id, first\_name, last\_name, address, city, phone, join\_date)

VALUES (member\_id\_seq.NEXTVAL, 'Mark', 'Quick-to-See',

'6921 King Way', 'Lagos', '63-559-7777', TO\_DATE('07-APR-1990', 'DD-MM-YYYY'))

/

INSERT INTO member(member\_id, first\_name, last\_name, address, city, phone, join\_date)

VALUES (member\_id\_seq.NEXTVAL, 'Audry', 'Ropeburn',

'86 Chu Street', 'Hong Kong', '41-559-87', TO\_DATE('18-JAN-1991', 'DD-MM-YYYY'))

/

INSERT INTO member(member\_id, first\_name, last\_name, address, city, phone, join\_date)

VALUES (member\_id\_seq.NEXTVAL, 'Molly', 'Urguhart',

'3035 Laurier', 'Quebec', '418-542-9988', TO\_DATE('18-JAN-1991', 'DD-MM-YYYY'));

/

COMMIT

SET VERIFY ON

1. Add the following movie copies in the TITLE\_COPY table: **Note:** Have theTITLE\_IDnumbers available for this exercise.

INSERT INTO title\_copy(copy\_id, title\_id, status)

VALUES (1, 92, 'AVAILABLE')

/

INSERT INTO title\_copy(copy\_id, title\_id, status)

VALUES (1, 93, 'AVAILABLE')

/

INSERT INTO title\_copy(copy\_id, title\_id, status)

VALUES (2, 93, 'RENTED')

/

INSERT INTO title\_copy(copy\_id, title\_id, status)

VALUES (1, 94, 'AVAILABLE')

/

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices: Solutions - 10**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices: Case Study Solutions (continued)**

INSERT INTO title\_copy(copy\_id, title\_id, status)

VALUES (1, 95, 'AVAILABLE')

/

INSERT INTO title\_copy(copy\_id, title\_id,status)

VALUES (2, 95, 'AVAILABLE')

/

INSERT INTO title\_copy(copy\_id, title\_id,status)

VALUES (3, 95, 'RENTED')

/

INSERT INTO title\_copy(copy\_id, title\_id,status)

VALUES (1, 96, 'AVAILABLE')

/

INSERT INTO title\_copy(copy\_id, title\_id,status)

VALUES (1, 97, 'AVAILABLE')

/

1. Add the following rentals to the RENTAL table:

**Note:** The title number may be different depending on the sequence number.

INSERT INTO rental(title\_id, copy\_id, member\_id,

book\_date, exp\_ret\_date, act\_ret\_date)

VALUES (92, 1, 101, sysdate-3, sysdate-1, sysdate-2)

/

INSERT INTO rental(title\_id, copy\_id, member\_id,

book\_date, exp\_ret\_date, act\_ret\_date)

VALUES (93, 2, 101, sysdate-1, sysdate-1, NULL)

/

INSERT INTO rental(title\_id, copy\_id, member\_id,

book\_date, exp\_ret\_date, act\_ret\_date)

VALUES (95, 3, 102, sysdate-2, sysdate, NULL)

/

INSERT INTO rental(title\_id, copy\_id, member\_id,

book\_date, exp\_ret\_date,act\_ret\_date)

VALUES (97, 1, 106, sysdate-4, sysdate-2, sysdate-2)

/

COMMIT

/

1. Create a view named TITLE\_AVAIL to show the movie titles, the availability of

each copy, and its expected return date if rented. Query all rows from the view. Order the results by title.

**Note:** Your results may be different.

|  |  |  |  |
| --- | --- | --- | --- |
| CREATE | VIEW title\_avail | | AS |
| SELECT | | t.title, c.copy\_id, c.status, r.exp\_ret\_date | |
| FROM |  | title t JOIN | title\_copy c |
| ON |  | t.title\_id = | c.title\_id |
| FULL | OUTER JOIN rental r | | |
| ON |  | c.copy\_id = r.copy\_id | |
| AND |  | c.title\_id = | r.title\_id; |
|  |  |  |  |

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices: Solutions - 11**

**Additional Practices: Case Study Solutions (continued)**

SELECT \*

FROM title\_avail

ORDER BY title, copy\_id;

1. Make changes to the data in the tables.
   1. Add a new title. The movie is “Interstellar Wars,” which is rated PG and classified as a science fiction movie. The release date is 07-JUL-77. The description is “Futuristic interstellar action movie. Can the rebels save the humans from the evil empire?” Be sure to add a title copy record for two copies.

INSERT INTO title(title\_id, title, description, rating, category, release\_date)

VALUES (title\_id\_seq.NEXTVAL, 'Interstellar Wars', 'Futuristic interstellar action movie. Can the

rebels save the humans from the evil empire?', 'PG', 'SCIFI', '07-JUL-77')

/

INSERT INTO title\_copy (copy\_id, title\_id, status)

VALUES (1, 98, 'AVAILABLE')

/

INSERT INTO title\_copy (copy\_id, title\_id, status)

VALUES (2, 98, 'AVAILABLE')

/

1. Enter two reservations. One reservation is for Carmen Velasquez, who wants to rent “Interstellar Wars.” The other is for Mark Quick-to-See, who wants to rent “Soda Gang.”

INSERT INTO reservation (res\_date, member\_id, title\_id)

VALUES (SYSDATE, 101, 98)

/

INSERT INTO reservation (res\_date, member\_id, title\_id)

VALUES (SYSDATE, 104, 97)

/

7. Make a modification to one of the tables.

1. Run the script in lab\_apcs\_7a.sql to add a PRICE column to the TITLE table to record the purchase price of the video. Verify your modifications.

ALTER TABLE title

ADD (price NUMBER(8,2));

DESCRIBE title

1. Create a script named lab\_apcs\_7b.sql that contains UPDATE statements that update each video with a price according to the list provided. Run the commands in the script.

**Note:** Have theTITLE\_IDnumbers available for this exercise.

SET ECHO OFF

SET VERIFY OFF

UPDATE title

SET price = &price

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices: Solutions - 12**

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Additional Practices: Case Study Solutions (continued)**

WHERE title\_id = &title\_id;

SET VERIFY OFF

SET ECHO OFF

1. Create a report that contains each customer’s history of renting videos. Be sure to include the customer name, movie rented, dates of the rental, and duration of rentals. Total the number of rentals for all customers for the reporting period. Save the commands that generate the report in a script file named lab\_apcs\_8.sql*.*

**Note:** Your results may be different.

SET ECHO OFF

SET VERIFY OFF

|  |  |  |  |
| --- | --- | --- | --- |
| SELECT | m.first\_name||' | | '||m.last\_name MEMBER, t.title, |
|  | r.book\_date, | r.act\_ret\_date - r.book\_date DURATION | |
| FROM | member m, title | | t, rental r |
| WHERE | r.member\_id = m.member\_id | | |
| AND | r.title\_id = | t.title\_id | |

ORDER BY member;

SET VERIFY ON

SET ECHO ON

|  |
| --- |
| Oracle Internal & Oracle Academy Use Only |

**Oracle Database 10*g*: SQL Fundamentals I Additional Practices: Solutions - 13**

Oracle Internal & Oracle Academy Use Only