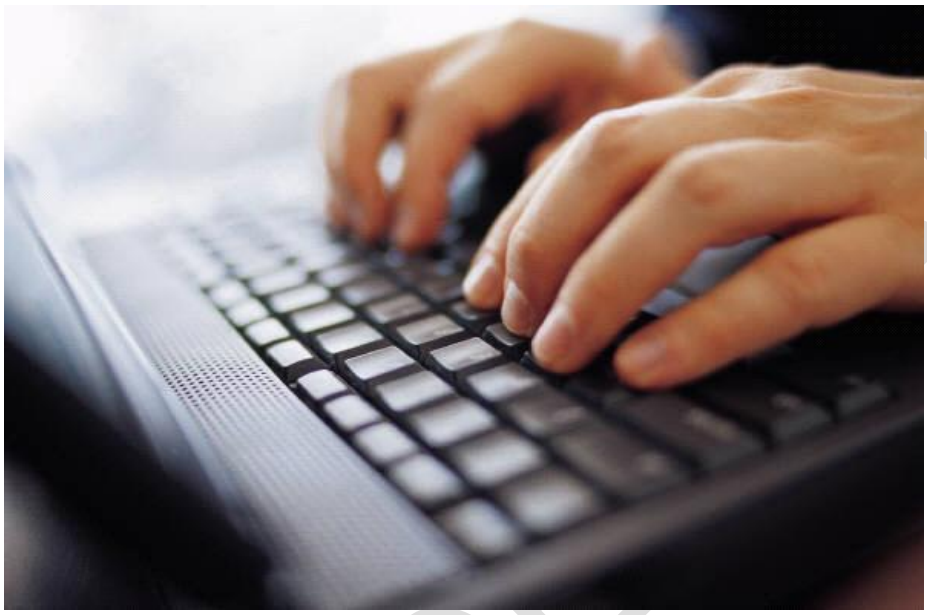
Lab Guide for RDBMS Essentials



Education and Research

Infosys®
POWERED BY INTELLECT
DRIVEN BY VALUES

CONFIDENTIAL

Author(s)	Rengarajan Ramanujam, Shinu Thomas
Authorized by	Srikantan Moorthy
Creation/Revision Date	June 2010
Version	1.4

COPYRIGHT NOTICE

All ideas and information contained in this document are the intellectual property of Education and Research Department, Infosys Technologies Limited. This document is not for general distribution and is meant for use only for the person they are specifically issued to. This document shall not be loaned to anyone, within or outside Infosys, including its customers. Copying or unauthorized distribution of this document, in any form or means including electronic, mechanical, photocopying or otherwise is illegal.

Education and Research Department
Infosys Technologies Limited
Electronic City
Hosur Road
Bangalore - 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infy.com
mailto:E&R@infy.com

CONFIDENTIAL

Document Revision History

Version	Date	Author(s)	Reviewer(s)	Comments
1.0	16-Mar-09	Rengarajan, Shinu Thomas	Kiran RK	Initial Draft
1.1	12-May-09	Rengarajan, Shinu Thomas	Kiran RK	Review comments incorporated after Pilot
1.2	15-May-09	Rengarajan, Shinu Thomas	Kiran RK	Review comments incorporated after Pilot
1.3	11-Mar-10	Shinu Thomas	Rengarajan	IP correction incorporated
1.4	30-Jun-2010	Shinu Thomas	Rengarajan	Checked for Devsquare compatibility

CONFIDENTIAL

Contents

CONFIDENTIAL

1 Background

This document contains assignments to be completed as part of the hands on for the subject RDBMS Essentials **(Course code: DB91).**

Note: In order to complete the course, assignments in this document must be completed in the sequence mentioned.

1 Assignments for Day 5 of RDBMS Essentials

NOTE: Please SET the AUTOCOMMIT to OFF

1.1 Demo 1: LOCKS

Objective: To be able to understand and appreciate the concept of exclusive lock.

Problem Description:

Different locks and the mechanisms are dealt during the class room session. Let us see practically the effect of exclusive locking.
For our demo, let us use Branch table.

Note: In Oracle the default locking is Intent Exclusive (row exclusive in Oracle terms) lock for Update/Delete/Insert

Step 1: Open two instances of SQL * PLUS on your computer system and login with your database credentials in both the instances.

Step 2: Go to first instance.
Write the following query:

```
UPDATE branch SET seatsavailable=8 WHERE branchid = 'B2';
```

Step 3: Now switch to the second instance.
Write the following query:

```
UPDATE branch SET seatsavailable=8 WHERE branchid = 'B2';
```

CONFIDENTIAL

Note: The second instance is hanged. The reason is that, the first transaction has issued the UPDATE statement for the branchid 'B2' and the table acquires the Intent exclusive (IX) lock and the record acquires the exclusive(X) lock. The second transaction is also trying to update the same record but because of the X lock acquired by the first transaction it has to wait.

Step 4: Go to the first instance and type COMMIT or ROLLBACK.

Step 5: Now switch to the second instance. You get the SQL prompt.

Note: As soon as you Commit/Rollback the transaction in second instance the X lock is released.

Step 6: Now type COMMIT or ROLLBACK in the second instance as well.

Summary of this assignment:
You have just learnt

- Update/Delete/Insert DML statement acquire a row exclusive lock
- The X lock is released only at the end of transaction which is marked by Commit/Rollback.

1.2 Demo 2: Deadlock

Objective: To be able to recognize when and how the deadlock occurs.

Problem Description:
To comprehend the problem of deadlock.

Estimated time: 10 minutes

Step 1: Open two instances of SQL * PLUS (If you have closed the previous ones) on your computer system and login with your database credentials in both the instances.

Step 2: Go to first instance.
Write the following query:

UPDATE branch SET seatsavailable=9 WHERE branchid= 'B1';

Step 3: Now switch to the second instance.
Write the following query:

CONFIDENTIAL

UPDATE branch SET seatsavailable=10 WHERE branchid= ‘B2’;

Step 4: Go to first instance.
Write the following query:

UPDATE branch SET seatsavailable=10 WHERE branchid= ‘B2’;

Step 5: Go to the second instance.
Write the following query:

UPDATE branch SET seatsavailable=9 WHERE branchid= ‘B1’;

Note: This situation is called the DEADLOCK. The Oracle is smart enough to detect it and it rolls back the statement of Step 4. Now check the Oracle message:
ORA-00060: deadlock detected while waiting for resource

Step 6: Issue COMMIT/ROLLBACK in both the instances.

Summary of this assignment:

We have just learnt
- How to visualize the deadlock.
- Oracle intelligently determines the deadlock situation

1.3 Assignment 1: Developer’s Corner

Objective: To be able to create or formulate a proper PL/SQL code when problem statement is provided.

Problem Description:

Consider the table Applicant.
Arrange the below code fragments (pieces of codes) provided in the appropriate order in order to implement each one of the business logics mentioned below. Also remember that repeated use of the same code fragment is also allowed.

1. Construct a PL/SQL block which would accept the applicant id as input and displays the applicant name only for valid applicant id.
2. Formulate a PL/SQL block which accepts the applicant id as input and retrieves the entire applicant record from the applicant table and later displays the applicant name and opted branch id of the given applicant.

CONFIDENTIAL

3. Develop a PL/SQL block which has to display the applicant name and the over all percentage of the applicant for a given applicant.
4. Formulate a PL/SQL block which accepts the applicant number as input and displays “No such applicant with this applicant id”, if the given applicant id is invalid else displays the applicant name and over all percentage of that applicant.
5. Construct a PL/SQL block which would display applicant name and branch id of the given applicant only in the case of valid applicant id else throw an appropriate error message, “No such applicant with this applicant id”.
6. Develop a PL/SQL block to display the count of total number of applicants opted for the given branch.
7. Develop a PL/SQL block to identify whether the given applicant opted for the branch id B3 and if his over all percentage is greater than or equal to 70. If so, print the applicant name and his overall percentage else throw an error message “No such applicant with this applicant id”.

(a) v_applicantid applicant.applicantid%TYPE;
(b) v_count NUMBER;
(c) v_applicantname applicant.applicantname%TYPE;
(d) v_overallpercentage applicant.overallpercentage%TYPE;
(e) rec_applicant applicant%ROWTYPE;
(f) v_optedbranch applicant.optedbranch%TYPE;
(g) v_overallpercentage applicant.overallpercentage%TYPE;
(h) DECLARE
(i) BEGIN
(j) v_applicantid:=&v_applicantid;
(k) SELECT applicantname INTO v_applicantname FROM applicant WHERE applicantid=v_applicantid;
(l) SELECT * INTO rec_applicant FROM applicant WHERE applicantid=v_applicantid;
(m) SELECT applicantname, overallpercentage INTO v_applicantname, v_overallpercentage FROM applicant WHERE applicantid=v_applicantid;
(n) IF v_optedbranch = ‘B3’ THEN
END IF;
(o) DBMS_OUTPUT.PUT_LINE(‘Applicant name : ‘|| v_applicantname);
(p) DBMS_OUTPUT.PUT_LINE(‘Overall percentage :’||v_overallpercentage);
(q) SELECT count(*) INTO v_count WHERE optedbranch=v_optedbranch;
(r) DBMS_OUTPUT.PUT_LINE(‘Applicant Opted branch:’||rec_applicant.optedbranch);
(s) IF v_overallpercentage >= 70 THEN

CONFIDENTIAL

END IF;

(t) DBMS_OUTPUT.PUT_LINE('Applicant name: '||rec_applicant.applicantname);

(u) DBMS_OUTPUT.PUT_LINE('No such applicant with this applicant id');

(v) SELECT applicantname, overallpercentage INTO v_applicantname, v_overallpercentage FROM applicant WHERE applicantid=v_applicantid;

(w)IF v_count = 0 THEN

ELSE

END IF;

(x) SELECT COUNT(applicantid) INTO v_count FROM applicant WHERE applicantid=v_applicantid;

(y) END;

(z) v_branchid:=rec_applicant.optedbranch;

(aa) v_optedbranch:=&v_optedbranch;

1.4 Case study based assignments:

Objective: To be able to formulate a proper PL/SQL code when problem statement is provided.

Details: 3 assignments have to be solved as a part of given case study. Check your batch portal for the details related to case study as well as assignments related to them.

CONFIDENTIAL