

# ASSIGNMENT #4 – Database Objects, Concurrency Control

ISQA 4300/8306

Fall 2010

**Due date:      Wednesday, November 10**

## Overview

We have been studying a variety of database objects (several kinds of tables, indexes) and associated details of storage and management. This assignment is designed to give you experience working with these concepts and observing their behavior in practice. In particular, you will

- Create partitioned tables
- Analyze tables
- Create b-tree, bitmap, function-based, and/or reverse key indexes
- Modify indexes
- Analyze indexes
- In the last part of the assignment, you will explore in depth the behavior of various row-level and table-level locks.

## Task 1: Working with Tables (25%)

Create four tables, called CUSTOMER1, CUSTOMER2, CUSTOMER3, and CUSTOMER4, having the same columns and data as PWOLTP.CUSTOMER. CUSTOMER1 will be a normal table, stored in tablespace USERS. HOWEVER, you will create CUSTOMER2, CUSTOMER3, and CUSTOMER4 as partitioned tables. Create these three tables, each with a different form of partitioning: range partitioning, hash partitioning, and list partitioning. For each, use two partitions, one placed in the USERS tablespace, and one placed in one of the tablespaces you created in the last assignment. For the range partitioned table, partition based on whether cust\_credit\_limit is < or >= 10,000. For the Hash partitioned, table, use cust\_credit\_limit as the basis for the hash. For the list partitioning, partition based on the values in the cust\_credit\_limit column. Place rows with cust\_credit\_limit values of 7000 & 9000 in one partition; place rows with the other cust\_credit\_limit values in another.

The tables should all end up with all of the data from the PWOLTP.CUSTOMER table. That is, do more than just create the tables; populate them as well.

Hint: You can get a head-start on the CREATE TABLE syntax by viewing the PWOLTP.CUSTOMER table in SQL Developer (click on the table name) and then viewing the SQL tab. You do not need to create any of the indexes/constraints.

**Q1:** Show the DDL statements for the four tables. Note any error messages encountered and explain how you corrected them.

**Q2:** Analyze the four CUSTOMERx tables and compare and contrast them and their underlying storage structures (In what ways are they similar? In what ways are they different?) I am deliberately leaving the comparison criteria vague. I want you to investigate the ways the tables are similar and different. The more observations the better.

**Q3** (answer before completing Task 2): Which of the types of partitioning do you think would be best for this table, given the type of query shown in task 2? Why?

## Task 2: Using Indexes (35%)

### *Subtask 1*

When you created the tables in Assignment 3, we did not think much about the indexes or where they would be located.

**Q4:** In which tablespace did your indexes get created? Why?

It is good practice to have a tablespace dedicated to indexes that is separate from the tablespaces used to store tables. Relocate the indexes created on your Diveshop tables in Assignment 3 to the INDX tablespace.

**Q5:** record the SQL statements used to relocate your indexes.

### *Subtask 2*

You have learned the concepts behind indexing and examined a number of different types of indexes. Now, you have the opportunity to apply these concepts. You will be given a query, shown below. Your task is to improve the performance of this query as much as possible, just by judicious use of indexes.

Setup:

Execute the following SQL commands.

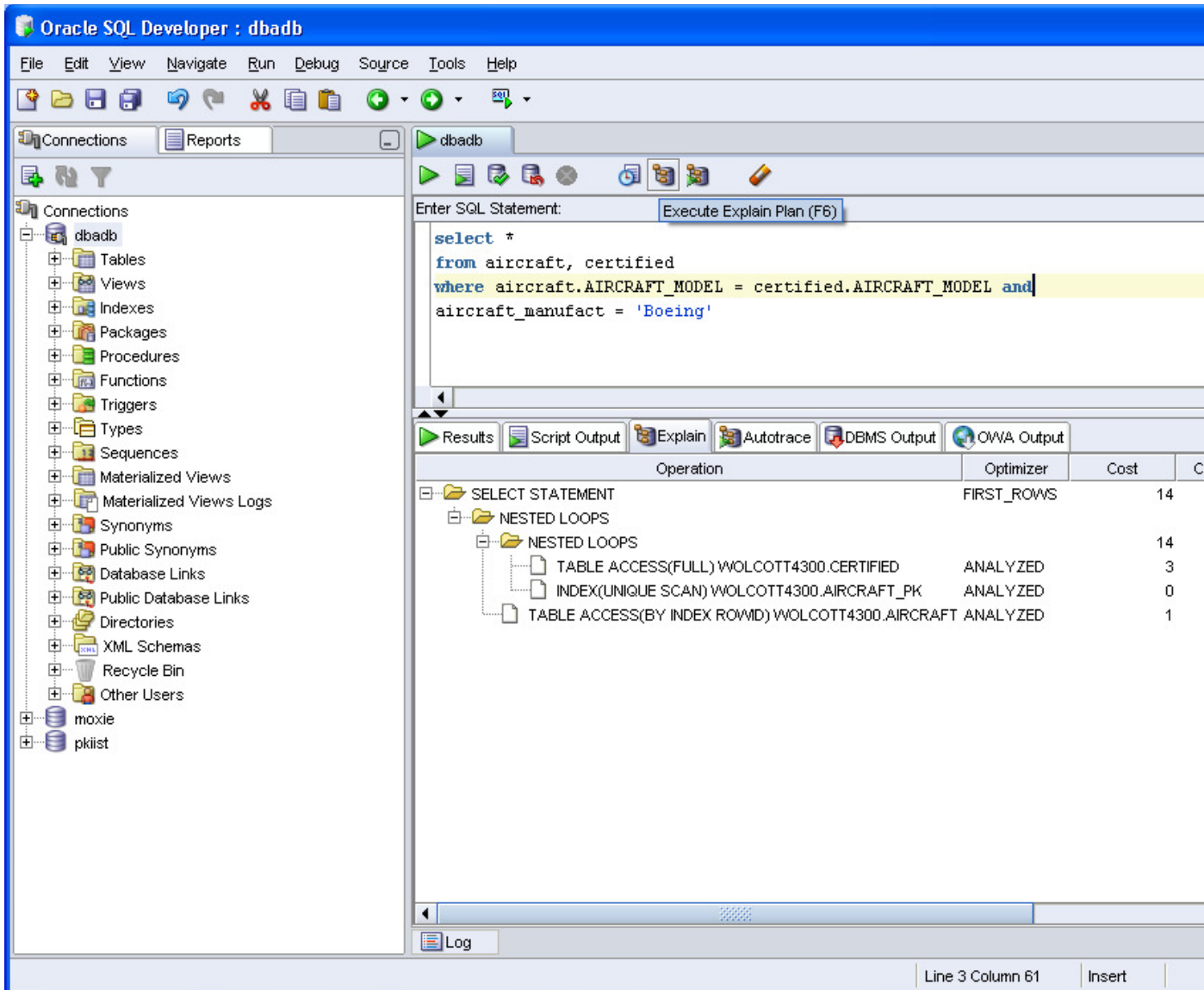
```
DROP TABLE CUSTOMER_PW;  
CREATE TABLE customer_pw AS  
SELECT * FROM PWOLTP.CUSTOMER;  
  
DROP TABLE CHANNEL;  
create table channel as  
select * from pwoltp.channel;  
  
DROP TABLE PRODUCT;  
Create table product as  
Select * from pwoltp.product;  
  
DROP TABLE ORDER_DETAIL;  
Create table order_detail as  
Select * from pwoltp.order_detail;
```

```
DROP TABLE ORDERS;  
CREATE TABLE orders AS  
SELECT * FROM PWOLTP.ORDERS;  
  
COMMIT;
```

The query you will work with is:

```
SELECT chan.channel_desc, c.cust_credit_limit, count(*)  
"No. Orders"  
FROM (SELECT *  
      FROM customer_pw  
      WHERE cust_credit_limit = 7000 OR  
            cust_credit_limit = 9000  
      ) c,  
orders o,  
order_detail od,  
product p,  
channel chan  
WHERE c.cust_id = o.cust_id AND  
      o.channel_id = chan.channel_id AND  
      o.order_id = od.order_id AND  
      od.prod_id = p.prod_id AND  
      p.prod_status = 'not available' AND  
      p.subcat_name = 'Jeans - Men' AND  
      TO_NUMBER(TO_CHAR(time_id, 'YYYY')) = 2000  
GROUP BY chan.channel_desc, c.cust_credit_limit  
ORDER BY chan.channel_desc, c.cust_credit_limit;
```

To analyze the performance of the query, you will use SQL Developer's ability to execute an "explain plan" and see the "cost" of the query. For example, in the screen shot below, I typed in the query, then clicked on the 'execute explain plan' icon, and then could see the explain plan. We will cover in more detail the explain plan later, but the important point for this exercise is that the "cost" is a measure of the amount of time/resources needed to execute the query. The lower the "cost" the better. The value of cost for the whole query is value in the top line of the explain plan (SELECT STATEMENT), 14 in this case.



**Q6.** Show the screen shot of the first time you ran the query. This will show the cost of the initial query.

**Q7.** Try substituting into the query the various partitioned versions of the CUSTOMER table. Does partitioning the CUSTOMER table in this case affect the performance of the query? Why or why not? Do your results confirm or disprove the hypothesis you made above in Task 1, Q3?

**Q8.** List the SQL for each index you create that contributes to your final answer.

**Q9.** Show a screen shot of the final run of your query.

Be sure you have an index created on some column within the CUSTOMER\_PW table. Analyze the index, as shown on slide #77 (p.22 of handout) in the lecture on indexes.

**Q10.** How many blocks does the index use?

**Q11.** How many leaf rows are there?

**Q12.** How many deleted leaf rows are there?

Drop half of the rows in the CUSTOMER\_PW table. Analyze the index again.

**Q13.** On examining the index\_stats table again, what has changed? What has not changed?

Rebuild the index (use the REBUILD command), placing it in the third tablespace you created for assignment #3. Analyze the index again.

**Q14:** List the query you used to rebuild the index

**Q15.** On examining the index\_stats table again, what has changed? What has not changed?

### Task 3: Locking & Concurrency Control (35%)

For this task, you will need to use SQL\*Plus. SQL Developer will not show the necessary behavior & messages.

1. Log into Oracle and execute the following statements:

```
CREATE TABLE EMPLOYEE AS SELECT * FROM WOLCOTT.EMPLOYEE;
```

2. Log into Oracle again, creating two simultaneous sessions. You may wish to change the SQL> prompt for the sessions so that you can easily distinguish between the two as you do this exercise. For example,

```
SQL> set sqlprompt "Session 1> "
```

3. For each row in the table below, execute the query shown in either session #1 or session #2, as indicated. Fill in the Row Lock, Table Lock, and Explanation columns as shown in the example of the first six rows. The objective is to indicate the locking behavior of the sessions and provide an explanation of this behavior.

**NOTE:** When a lock is requested by a session, but not granted, put the name of the lock in parentheses (), as shown in row #2 below.

Step	Session #1	Session #2	Row Lock	Table Lock	Explanation
2	LOCK TABLE employee IN ROW SHARE MODE;		n/a	RS lock	Session 1 acquires requested table lock.

3		LOCK TABLE employee IN EXCLUSIVE MODE NOWAIT;	n/a	(X) (exclusive)	ORA-00054 error. Table Exclusive lock conflicts with RS lock.
4		SELECT emp_salary FROM employee WHERE emp_no = 7521 FOR UPDATE OF emp_salary;	X	RX lock (intent update)	Successful. RS lock is compatible with RS lock.
5	UPDATE employee SET emp_salary = emp_salary * 1.10 WHERE emp_no = 7521;		(X)	(RX) (intent exclusive)	Session #1 waits for lock, since Session #2 holds X lock on that particular row
6		ROLLBACK;	Session #1 granted X	Session #2 RX lock released; Session #1 granted RX	When session #2 locks are released, session #1 receives locks it had been waiting for.
7	ROLLBACK;				Session #1 RS, RX, X locks released
8	LOCK TABLE employee IN ROW EXCLUSIVE MODE;				
9		LOCK TABLE employee IN EXCLUSIVE MODE NOWAIT;			
10		LOCK TABLE employee IN SHARE ROW EXCLUSIVE MODE NOWAIT;			



11		UPDATE employee SET emp_salary = emp_salary * 1.1 WHERE emp_no = 7521;			
12		ROLLBACK;			
13	SELECT emp_salary FROM employee WHERE emp_no = 7521 FOR UPDATE OF emp_salary;				
14		UPDATE employee SET emp_salary = emp_salary * 1.1 WHERE emp_no = 7521;			
15	ROLLBACK;				
16		ROLLBACK;			
17	LOCK TABLE employee IN SHARE MODE;				
18		LOCK TABLE employee IN EXCLUSIVE MODE NOWAIT;			
19		LOCK TABLE employee IN SHARE ROW EXCLUSIVE MODE NOWAIT;			
20		LOCK TABLE employee IN SHARE MODE;			
21		SELECT emp_salary FROM employee WHERE emp_no = 7521;			

22		SELECT emp_salary FROM employee WHERE emp_no = 7521 FOR UPDATE OF emp_salary NOWAIT;			
23		UPDATE employee SET emp_salary = 2000 WHERE emp_no = 7521;			
24	ROLLBACK;				
25		ROLLBACK;			
26	LOCK TABLE employee IN SHARE ROW EXCLUSIVE MODE;				
27		LOCK TABLE employee IN EXCLUSIVE MODE NOWAIT;			
28		LOCK TABLE employee IN SHARE ROW EXCLUSIVE MODE NOWAIT;			
29		LOCK TABLE employee IN SHARE MODE NOWAIT;			
30		SELECT emp_salary FROM employee WHERE emp_no = 7521;			
31		SELECT emp_salary FROM employee WHERE emp_no = 7521 FOR UPDATE OF emp_salary NOWAIT;			

32		UPDATE employee SET emp_salary = 1500 WHERE emp_no = 7521;			
33	UPDATE employee SET emp_salary = 2000 WHERE emp_no = 7521;				
34	ROLLBACK;				
35		ROLLBACK;			
36	LOCK TABLE employee IN EXCLUSIVE MODE;				
37		LOCK TABLE employee IN EXCLUSIVE MODE NOWAIT;			
38		LOCK TABLE employee IN ROW EXCLUSIVE MODE NOWAIT;			
39		LOCK TABLE employee IN SHARE MODE NOWAIT;			
40		LOCK TABLE employee IN ROW SHARE MODE NOWAIT;			
41		SELECT emp_salary FROM employee WHERE emp_no = 7521;			
42		SELECT emp_salary FROM employee WHERE emp_no = 7521 FOR UPDATE OF emp_salary;			

43	UPDATE employee SET emp_salary = emp_salary * 1.2 WHERE emp_no = 7521;				
44	COMMIT;				
45	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;				
46	SELECT emp_salary FROM employee WHERE emp_no = 7521; <sup>1</sup>				
47		UPDATE employee SET emp_salary = 2500 WHERE emp_no = 7521;			
48		SELECT emp_salary FROM employee WHERE emp_no = 7521;			
49	SELECT emp_salary FROM employee WHERE emp_no = 7521;				
50		COMMIT;			
51	SELECT emp_salary FROM employee WHERE emp_no = 7521;				
52	COMMIT;				
53	SELECT emp_salary FROM employee WHERE emp_no = 7521;				

---

<sup>1</sup> In this and the following SELECT states, please include in your answer the value returned by the query.

## For ISQA 8306 students only (15%):

One of the questions we have not answered in class is what happens when a transaction requests multiple types of locks on the same table. Does the transaction hold multiple locks simultaneously? Does Oracle maintain only the most restrictive lock? Does some other phenomenon happen? Does the order in which the locks are requested matter?

Through experimentation, explore Oracle's management of table locks. You will need to use either the v\$lock data dictionary view, or Enterprise Manager's instance locks page. You will need to do sufficient experimentation that you can convince me that you know the answers to the questions.

Please turn in the SQL you used for your experimentation, and screen shots of the results. In a single paragraph, summarize your conclusions.

## Task 4: Reflection (5%)

Pick **two** (2) of the bullet points from this list and answer the questions<sup>2</sup>. Please indicate which questions you are answering.

- What did I actually achieve with this piece of work? Which were the most difficult parts, and why were they difficult for me? Which were the most straightforward parts, and why did I find these easy?
- How well do I think I achieved the intended learning outcomes for this task? Where could I have improved my achievement? Why didn't I improve it at the time?
- What have I got out of doing this assignment? How have I developed my knowledge and skills? How do I see the payoff from doing this assignment helping me in the longer term?
- What else have I got out of doing this assignment? Have I developed other skills and knowledge, which may be useful elsewhere at another time? If so, what are my own emergent learning outcomes from doing this assignment?
- What was the best thing I did? Why was this the best thing I did? How do I know that this was the best thing I did?
- What worked least well for me? Why did this not work well for me? What have I learned about the topic concerned from this not having worked well for me? What have I learned about myself from this not having worked well for me? What do I plan to do differently in future as a result of my answers to the above questions?
- With hindsight, how would I go about this assignment differently if doing it again from scratch? To what extent will this assignment influence the way I tackle anything similar in future?

---

<sup>2</sup> Questions taken from <http://www.escalate.ac.uk/resources/reflection/09.html>

- What did I find the greatest challenge in doing this work? Why was this a challenge to me? To what extent do I feel I have met this challenge? What can I do to improve my performance when next meeting this particular sort of challenge?
- What was the most boring or tedious part of doing this assignment for me? Can I see the point of doing these things? If not, how could the assignment have been re-designed to be more stimulating and interesting for me?
- Has it been worth the effort I put in? Do the marks represent a just reward? Should this assignment be worth more or less marks in the overall scheme of things?
- Do I feel that my time on this assignment has been well spent? If not, how could I have used my time more sensibly? Or should the assignment have been designed differently? Which parts of the assignment represent the time best spent? Which parts could be thought of as time wasted?
- How useful do I expect the feedback to be, that I receive on my efforts for this assignment? What sorts of feedback do I really want at this point in time? What sorts of feedback do I really need at this point in time? What are my expectations of getting useful feedback now, based on the feedback (or lack of it) that I've already received on past work?
- Overall, how has this assignment helped (or hindered) my motivation to learn more about this part of my syllabus? Has it encouraged me, or disillusioned me?
- To what extent has this assignment helped me to clarify what I need to learn about this topic? Have I a clearer picture after doing the assignment, or a foggier one? Who can help me gain a clearer picture, if the latter?
- To what extent has this assignment helped me to see where the goalposts stand for future assessments such as exams? Has it given me useful insights into what will be expected of me in future?
- What advice would I give go a friend about to start on the same assignment? How much time would I suggest that it would be worth putting into it? What pitfalls would I advise to be well worth not falling into?
- What are the three most important things that I think I need to do with this topic at this moment in time? Which of these do I think is the most urgent for me to do? When will I aim to start doing this, and what is a sensible deadline for me to have completed it by?

## **Deliverables**

### **Task 1 & 2**

- Provide the answers/screen shots requested for question Q1-Q12. Please note that I reserve the right to include my own ability to reproduce your query costs into the evaluation of your deliverable. If I can't obtain results that are

reasonably close to yours by going through the same steps as you, then I will adjust your grade accordingly.

#### Task 3

- I recommend copying the entire table from Task 3, pasting it into your solution document, and filling in the answers in the table itself.

#### Task 4

- Answers to questions in two bullet points (your pick).