

Homework Assignment 1

(due through Blackboard before 6:00 PM on Wednesday, 1/24)

Things you should please do, lest you should lose my tender affection:

- Review the syllabus rules for submitting assignments
- Somehow incorporate your last name into the name of the file you upload to the Dropbox
- Make sure your name is inside the document itself, at the beginning or in the header

Certification of Academic Honesty (see syllabus for details)

=====

I certify that:

- This homework assignment is entirely my own work.
- I have not quoted the words of any other person from a printed source or a website without indicating what has been quoted and providing an appropriate citation.
- I have not submitted this paper / project to satisfy the requirements of any other course.

Signature: Joseph Mulray
Date : 1/23/18

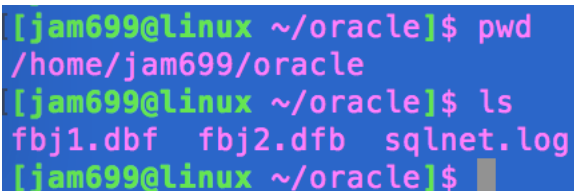
=====

Problem 1: Where's My Table? (40 points)

Step 1: Using Oracle SQL*Plus (the command line) or SQL Developer, create a new tablespace called "FunnyBonesJones" that is composed of two operating system files (put 'em wherever you want to put them, and name them whatever you like) – show the command you used to create the tablespace, and a screenshot of some operating system utility (e.g., Windows Explorer, dir) which demonstrates that the files have been created.

Answer:

```
CREATE TABLESPACE FunnyBonesJones
DATAFILE
'/home/jam699/oracle/fbj1.dbf',
'/home/jam699/oracle/fbj2.dbf';
```



```
[jam699@linux ~/oracle]$ pwd
/home/jam699/oracle
[jam699@linux ~/oracle]$ ls
fbj1.dbf  fbj2.dfb  sqlnet.log
[jam699@linux ~/oracle]$
```

Step 2: Sing the complete lyrics of “Wrecking Ball” from memory. Repeat until you have it just right. (If you don’t do this, I will know - the ashamed look in your eyes will give you away.)

Very easy.

Step 3: Create a table of any design you choose in your FunnyBonesJones tablespace. Name the table whatever you like. Show the command you used to create the table. (Don’t add any data.)

Answer:

```
CREATE TABLE FunnyBonesTable(  
ID number (1) PRIMARY KEY);
```

Step 4: Develop a single query (yes, one query – no fancy two-stepping here) to determine the file in which your table is located. DBA_EXTENTS and DBA_DATA_FILES are your friends. Joining these two views together would be a good plan. Note that the concept of a “segment” will be part of the solution. Show the query and your results. (Hint: If you are stuck on this, you can do a bit of Google sleuthing and find a demonstration of it on my YouTube channel.)

Answer:

```
SELECT DBA_DATA_FILES.FILE_NAME FROM DBA_DATA_FILES  
JOIN DBA_EXTENDS ON  
DBA_DATA_FILES.TABLESPACE_NAME = DBA_EXTENDS.TABLESPACE_NAME  
WHERE DBA_DATA_FILES.FILE_NAME LIKE "%TBS%";
```

Step 5: Get rid of the FunnyBonesJones tablespace, along with the table you created and any associated operating system files. Show the command you used.

Answer:

```
DROP TABLESPACE FunnyBonesJones  
INCLUDING CONTENTS AND DATAFILES;
```

Problem 2: Stifler Search (15 points)

Fred and Tina are both running working with their company’s Oracle database at the same time. Fred starts a long-running query on Table A that looks for customers with the last name of “Stifler.” Shortly thereafter, Tina issues a command to change the name of customer 1234, Janine Stifler, to Janine Finch. She commits her change before Fred’s query finishes. Do Fred’s results include customer 1234? (Briefly explain your reasoning.)

Answer:

No the result would not contain the commits made by Tina. Oracle enforces statement-level read consistency that guarantees that all data returned by a single query comes from a single point in time,

which is when the query began. Resulting in Fred not pulling Tina's updated changes despite having the query still running.

Problem 3: Index Types (15 points)

Imagine that you have a table of customers containing thousands of rows. What kind of index (bitmap or b-tree) would you use for the following columns? (Briefly explain your reasoning for each choice).

- a) CustomerId
- b) LastName
- c) MaritalStatus

Answer:

a)

CustomerId:

b-tree would be more suitable for a situation using CustomerId. There are several disadvantages for using a bitmap index on unique columns for example space.

b)

LastName:

b-tree would also be more suitable for a situation using LastName. Even though there is the possibility of multiple data fields that are the same there would still be more advantages of using a b-tree over bitmap in this case.

c)

bit-map is something I would use in this case of Marital Status that has low distinct values. Marriage is something that is not constantly being frequently updated and changes so a bit-map would be more suitable in a situation like this.

Problem 4: When I Say Fast, I Mean Garofalo Fast (30 points)

You get a breathless phone call from Janeane Garofalo, who tells you that she needs to improve the performance of a Java / Oracle application she has written to defeat the enemies of the United States. She send you a copy of her code by email. In your review of her application, you notice the following code snippets:

From GetTheBadGuys.java:

...

Frein – 1/18/2018

```
String sql = "SELECT LastName, Phone FROM Guys WHERE GuyType = 'BAD' AND LastName <> 'Stiller'";
```

```
ResultSet badGuys = stmt.executeQuery(sql);
```

...

From GetAnyStubbornlyRemainingBadGuys.java:

...

```
String sql = "SELECT LastName, Phone FROM Guys WHERE LastName <> 'Stiller' AND GuyType = 'BAD'";
```

```
ResultSet badGuys = stmt.executeQuery(sql);
```

...

Once again, fate has called upon you to help Janeane Garofalo save this nation. What recommendation do you make?

NOTE: You are not allowed to not help her. This is Janeane Garofalo we are talking about here. Have some respect, people.

Answer:

She is executing virtually the same query again with the same conditionals. She will have the same resultset from each of the queries despite the order of selecting LastName<>'Stiller' AND GuyType ='BAD'. Execute the query once if she only needs LastName, and Phone number then use the same dataset that was returned. No need to execute two queries separately.

<End of Assignment>

