**DBAS 1100 Assignment 3 – Self & Cross Joins, Set operations**

**\*Solutions at End**

**Value:** 10% of your overall mark.

**Due Dates: Second class, Week of April 13th, Before start of class**

**Required Database(s):**

* Chinook
* Bookstore
* Numbers
* Cars
* Lunches

**Instructions:**

Create a single script file containing SQL statements to meet the query requirements listed below.

For this assignment, you will be using five separate databases: **Chinook, Bookstore, Numbers, Cars** and **Lunches**. For the latter two, you will need to run the following DDL scripts to create the two databases required for this assignment.

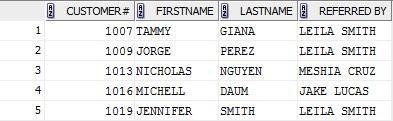
**Cars.sql** – Creates the “Cars\_” tables needed for this assignment.

**Lunches\_db.sql** – Creates the “L\_” tables needed for this assignment.

When you are finished, upload your script to D2L as your submission for Assignment 3, using a file name similar to: **[YourName]\_DBAS\_Assignment3.sql**.

**Query Requirements:**

1. **(Cars db)** Using the Cars\_Car\_Types, Cars\_Number\_Of\_Doors, Cars\_Colors tables, create a query that returns every possible combination of the values of each table. (Hint: The result set should contain 24 rows.)
2. **(Lunches db)** List the employee ID, last name, and phone number of each employee with the name and phone number of his or her manager. Make sure that every employee is listed, even those that do not have a manager. Sort the rows by the employee’s id number.
3. **(Bookstore db)** List the customer#, first name and last name of each customer that has been referred by another customer. Include the first name and last name of the customer who did the act of referring. Research a method on the internet to combine the first and last names of the referrer so that it displays in one column labeled “Referred By”. Make sure that there is a space between the first name and last name when you combine the two. The results should look **exactly** as follows:



1. **(Multiple dbs)** Create one full list of first names and last names of all customers from the Chinook tables, all authors from the Bookstore tables, all customers from the Bookstore tables, and all employees from the Lunches tables. Sort the list by last name and first name in ascending order.
2. **(Numbers db)** Using the Numbers\_Twos and Numbers\_Threes tables, show the results of a query that only displays numbers that do not have a matching value in the other table.
3. **(Numbers db)** Using the Numbers\_Twos and Numbers\_threes tables, show the results of a query that only displays numbers that have a matching value in the other table. Here’s the catch: You are not pe**r**mitted to use a WHERE clause or joins for this query.

(Hint: <http://docs.oracle.com/cd/B28359_01/server.111/b28286/queries004.htm>)

SOLUTIONS:

/\* DBAS 1100 - 700 ASSIGNMENT 3

CHRIS ARBUTHNOTT, W0276812

APRIL 13TH, 2015 \*/

--1. (CARS) List all type, number of doors, and color combinations possible.

SELECT \* FROM CARS\_COLORS COL

CROSS JOIN CARS\_CAR\_TYPES TYP

CROSS JOIN CARS\_NUMBER\_OF\_DOORS DOR;

--2. (LUNCHES) List all employee ids, last names and phone nums, along with their

-- manager's name and phone number.

SELECT EMP.EMPLOYEE\_ID, EMP.LAST\_NAME, EMP.PHONE\_NUMBER,

MAN.LAST\_NAME AS "Manager", MAN.PHONE\_NUMBER AS "Manager Phone Number"

FROM L\_EMPLOYEES EMP LEFT JOIN L\_EMPLOYEES MAN ON EMP.MANAGER\_ID=MAN.EMPLOYEE\_ID

ORDER BY EMP.EMPLOYEE\_ID;

--3. (BOOKSTORE) List the ids, first and last names of all customers that have

-- been referred. Include the name of referrer as a single field.

SELECT REFD.CUSTOMER#, REFD.FIRSTNAME, REFD.LASTNAME,

CUS.FIRSTNAME || ' ' || CUS.LASTNAME AS "REFERRED BY"

FROM B\_CUSTOMERS CUS INNER JOIN B\_CUSTOMERS REFD ON REFD.REFERRED=CUS.CUSTOMER#

ORDER BY REFD.CUSTOMER#;

--4. (CHINOOK, BOOKSTORE, LUNCHES) Compile a large list of first and last names

-- including customers from chinook, customers and authors from bookstore,

-- and employees from lunches.

SELECT \* FROM (

SELECT FIRSTNAME, LASTNAME FROM CUSTOMER

UNION

SELECT FNAME AS FIRSTNAME, LNAME AS LASTNAME FROM B\_AUTHOR

UNION

SELECT FIRSTNAME, LASTNAME FROM B\_CUSTOMERS

UNION

SELECT FIRST\_NAME AS FIRSTNAME, LAST\_NAME AS LASTNAME FROM L\_EMPLOYEES

)

ORDER BY LASTNAME, FIRSTNAME;

--5. (NUMBERS) Display the entries from numbers\_twos and numbers\_threes that

-- do \*not\* have entries in both tables.

(SELECT MULTIPLE\_OF\_2 AS "Number" FROM NUMBERS\_TWOS

UNION SELECT MULTIPLE\_OF\_3 AS "Number" FROM NUMBERS\_THREES)

MINUS

(SELECT MULTIPLE\_OF\_2 FROM NUMBERS\_TWOS

INTERSECT SELECT MULTIPLE\_OF\_3 FROM NUMBERS\_THREES

);

--STRATEGY: (THE UNION) MINUS (THE INTERSECTION)

--6. (NUMBERS) Display the entries from numbers\_twos and numbers\_threes that

-- \*do\* have entries in both tables.

SELECT MULTIPLE\_OF\_2 AS "Number" FROM NUMBERS\_TWOS

INTERSECT

SELECT MULTIPLE\_OF\_3 AS "Number" FROM NUMBERS\_THREES;

--STRATEGY: STRAIGHT UP INTERSECTION!

--END ASSIGNMENT