

# Lab 2: Intermediate Oracle and SQL

## (Same as Problem Set 1)

---

**Due:** A day before lecture #6

### PURPOSE

This lab uses the ['parcels'](#)\* and ['urisa'](#)\* database tables stored in Oracle to help you think relationally about data and improve your understanding of how SQL 'select' statements handle 'group by', joins, self-joins, and missing values. You will also begin to create some temporary tables.

### PART 1: SQL Plus Logon and Setup

In order to access the “parcels” and “urisa” database, you need to log on using one of two different Oracle accounts--the user names are 'parcels' and 'urisa'; and the passwords are the same - the one we gave you in the first lab. We use separate accounts for security reasons. Each account can access only a few small tables - those needed for these exercises.

For your information, the SQL 'create' statements used to create the 'parcels' tables are shown in the ['PARCELS Database Schema'](#)\* link from the ['SQL Notes'](#)\* section of the class Web pages. The SQL 'create' statements used for the 'urisa' tables are shown in the ['URISA Database Notes'](#)\* link.

*This lab will require you to create temporary tables. Since all of you are using the same userid, there's a chance that more than one of you tries to create tables with the same name. To avoid such problems, we emailed a unique tablename prefix in the form 'Tnn' to each of you. For more information, please consult the section entitled ['Creating unique temporary tables'](#)\* in the [SQL help](#)\* notes.*

Beginning next week, you will connect to Oracle using your own individual account with the same userid as your Athena-id. We'll give you an initial temporary password. Your instructor will tell you your temporary password. Please change this password when you first login using your Athena ID using the SQL command: 'password'. You will be prompted for your old password and then the new password twice. We recommend that you use a different password from your Athena or CRL passwords.

---

\* Kindly refer to the Labs section

## PART 2: Querying the 'Toy' Parcel Database (30 points)

Construct SQL statements for the following queries:

- a. (2 points) Which parcels had square footage greater than 10,000 square feet? List the PARCELID, PID, WPB, ZIP, LANDUSE, and SQFT columns, and sort by PARCELID in ascending order.
- b. (2 points) Which of these parcels in **part a** had land use code 'C'?
- c. (3 points) Which of these parcels in **part a** had land use code 'C' or 'E'?
- d. (3 points) Repeat the answer from **part c**, but this time list the name of the owner in the first column.
- e. (3 points) What is the total loss for each owner whose parcel(s) burned? List the owner and the total loss. Sort by the owner's names in ascending order.
- f. (3 points) How many parcels do each owner own and what total square footage of land do they own? Include the owner name, the owner ID number, the parcel count, and total square footage in your listing. Sort by the owner's names.
- g. (3 points) Modify your answer to **part f** above to exclude any parcels with square footage less than 30,000 square feet.
- h. (3 points) Modify your answer to **part g** above to exclude any owners that only own one qualifying parcel.
- i. (4 points) The following queries produce slightly different results. Briefly explain why the results make sense for each case. Explain the difference or similarity between COUNT(\*) and COUNT(tax).

```
1. SELECT COUNT(*), COUNT(tax), MIN(tax), MAX(tax),  
   AVG(landval)  
   FROM tax;
```

```
2. SELECT COUNT(*), COUNT(tax), MIN(tax), MAX(tax),  
   AVG(landval)  
   FROM tax  
   WHERE tax IS NOT NULL;
```

- j. (4 points) What is the total value (land value plus building value) per square foot of each parcel? Sort by this statistic in descending order. Include the parcel id, building value, land value, and the square footage in your query. Make sure that you avoid potential "divide by zero" errors and

that you include only rows with valid values for the square footage, building value, and land value. Your query should return 8 rows. However, the TAX table has 9 rows and the PARCELS table has 20 rows. How do you account for having only 8 rows in the result?

### **PART 3: Querying the URISA Proceedings Database (30 points)**

- a. (9 points) Which keywords were used at least 30 times? Sort by the count of appearances in descending order.
- b. (9 points) List the authors who wrote at least 4 papers. Sort by the paper count in descending order.
- c. (12 points) Find the paper number and title (first 40 characters) of all papers using **both** keyword 305 (LOCATIONAL/SPATIAL ANALYSIS) **and** 218 (LAND USE PLANNING). Sort by the titles. Be aware that you need a self-join to avoid getting no rows in your result!

### **PART 4: Additional Questions (40 points)**

- a. (12 points) We want to write a query that computes the total loss from fires for each owner (who had at least one parcel that was in a fire) where that owner's total loss is greater than \$100,000. As a first attempt, we come up with this query:

```
SELECT P.PARCELID, O.ONAME, SUM(F.ESTLOSS) TOTAL_LOSS
FROM PARCELS P, OWNERS O, FIRES F
WHERE P.ONUM = O.OWNERNUM
GROUP BY O.ONAME
HAVING SUM(F.ESTLOSS) > 1000000
ORDER BY O.ONAME;
```

However, this query does not run. Write a corrected version of this query that works and show its output. Explain what was broken and how you fixed it. *Hint: More than one thing is wrong.*

- b. (14 points) Which of the papers written in 1988 had three or more authors? Show the paper number and author count, and sort by the author count in descending order, then the paper number in descending order.
- c. (14 points) Among all these 1988 papers written by multiple authors, which ones use keywords containing the word 'GEOGRAPHIC' or 'LAND' (but not 'NEW ENGLAND'). Do *not* use keyword numbers when writing your query or queries. List the paper ID and author names for each such paper. Sort them by the paper ID, the author's last name, and the author's first name.

## PART 5: WHAT TO TURN IN

Instead of a lab assignment, the answers to this lab constitute Problem Set 1. Turn in your answers (both the SQL statements and the result of the queries) to the [Part 2 \(parcels\)](#), and [Part 3 \(URISA\)](#), and [Part 4 \(additional\)](#) queries. Save your answers as a plain text file. You can use any text editors to cut and paste the relevant questions and your answers from the lab exercise page and your Oracle window into your lab assignment output.

To make it easier for us to rerun your queries, please do **not** include line numbers in your SQL statements (SQL\*Plus will print these by default). To avoid this, write your query in an editor first, then copy-and-paste the SQL statement into SQL\*Plus. Then, copy-and-paste the answer in SQL\*Plus into your editor buffer. *Any answer that includes line numbers in the SQL statement will lose a point for that question.*

While not required, we encourage you to format your SQL statements on multiple lines to improve their readability. Use the examples we provide as a guide.

Be careful to sort your results as stated in the questions. You will lose some credit if the results are not sorted correctly.

We only need your answer in digital format. The problem set is due a day before **lecture #6**.

---