

# PRACTICE Examination

*This is last year's exam. Note that it was considered a bit too long and too hard. This year's will be shorter and will include a bit more on database design and less of the most technical stuff. Otherwise, it will have a similar format.*

---

## Instructions

This is an open-book, open-note examination. You are free to use the web (especially the class web pages and Oracle documentation) to help you. Like any examination, however, we expect your submission to reflect your work exclusively. Hence any communication via any means (e.g., voice, paper, e-mail, zephyr) with anyone other than the class instructors is prohibited.

The exam duration is **90 minutes**. All the tables needed for the exam questions are accessible in Oracle via SQL\*Plus using your personal Oracle account that we have been using for lab exercises and homework.

The exam is worth a total of 100 points.

An effective way for you to prepare your exam answers is to use the same familiar methods you have used for the lab assignments and homework. Create an ASCII text or HTML file in your Athena account to contain your answers. Name your exam ***username\_exam.txt*** if you use text format or ***username\_exam.html*** if you use HTML format. For your own security, we suggest that you save it in the "Private" subdirectory of your Athena account (i.e., the user **smith** might create the text file ***/mit/smith/Private/smith\_exam.txt***). Note that you need the capital "P"! We strongly recommend that you retain the file containing your answers until we return the graded exams to you.

- *Remember to turn in your exam (in plain ASCII text or HTML format) before leaving the room.*
- *Make sure to include your name and Athena user ID near the beginning of your file (failure to include this will cost you 5 points; you lose 2 points if you supply only one piece of information).*
- *Please confirm with an instructor that your exam has been received.*

To turn in your exam, use the `turnin` command with assignment 100. If your exam were named `username_exam.txt`, you would use this command to turn it in from the 11.521 window:

```
turnin -c 11.521 100 username_exam.txt
```

*Finally, don't spend all your time on one or two questions. Start by looking over all the questions to get a general sense of what, and how much, is being asked, and the grade points associated with each question. Then start work on your answers but move on to the next question if you've spent more than 10 minutes on any one of them.*

**Good luck!**

## Part I: Short Answer and 'Fix SQL' Questions (45 points)

### Question I-1.

The query below is intended to find all the parcels owned by Gerald Rappaport, but it fails with the error 'ORA-00904: invalid column name':

```
SELECT p.parcelid, p.add1, p.add2, p.zip
FROM parcels p, owners o
WHERE p.onum = o.ownernum
AND o.oname = "GERALD RAPPAPORT";
```

### Question I-1a (3 points).

Why doesn't the query do what is intended?

### Question I-1b (7 points).

Rewrite this query to produce the correct result. Show your SQL and its output.

---

### Question I-2.

Take a look at the query and its output below, which is intended to list information about the parcels that:

- are greater than 20,000 square feet **or** are in ward-precinct-block (**wpb**) 112
- had at least one fire

```
SELECT p.parcelid, p.wpb, p.sqft, f.fdate
FROM parcels p, fires f
WHERE p.parcelid = f.parcelid
AND p.sqft > 20000
OR p.wpb = 112
```

```
ORDER BY p.parcelid, p.sqft, f.fdate;
```

PARCELID	WPB	SQFT	FDATE
2	292	100000	02-AUG-88
2	292	100000	02-APR-89
3	112	5500	01-AUG-87
3	112	5500	02-AUG-88
3	112	5500	02-APR-89
3	112	5500	02-JUL-89
3	112	5500	26-JUL-89
3	112	5500	26-JUL-90
20	543	50000	02-JUL-89

9 rows selected.

Notice that the parcel with **parcelid** = 3 is listed six times. However, this parcel had only two fires.

**Question I-2a (3 points).**

Why doesn't the query do what is intended? (Note: You should mention a specific term that applies to an aspect of this problem.)

**Question I-2b (7 points).**

Rewrite the query to obtain the right result. Show your SQL and its output.

---

**Question I-3(a) (5 points).**

Consider the structure of the **keywords** table in the URISA database. Which of the normal forms that we have discussed in detail in class does this table violate? Why?

**Question I-3(b) (5 points).**

How would you redesign this database *with respect to the **keywords** table only* to correct this problem? (The other URISA tables, **authors**, **titles**, and **match**, should remain unchanged.) Make sure to describe the resulting keys and their relationships. You do **not** need to write SQL statements that implement your result, although you are welcome to do so to augment your discussion. A verbal description of the changes you would make is sufficient.

---

#### Question I-4.

Suppose the city is considering reducing the building values (stored in the **tax** table) of parcels that experienced fires by the amount of the total estimated losses from all the fires. First, we can make a copy of the **tax** table called **mytax** that will hold the new values:

```
CREATE TABLE mytax AS
  SELECT *
  FROM tax;
```

Table created.

We can test the impact of this change running this query:

```
SELECT t.parcelid, t.bldval, SUM(f.estloss) losses,
       t.bldval - SUM(f.estloss) new_bldval
  FROM mytax t, fires f
 WHERE t.parcelid = f.parcelid (+)
 GROUP BY t.parcelid, t.bldval
 ORDER BY t.parcelid;
```

PARCELID	BLDVAL	LOSSES	NEW_BLDVAL
1	1600000		
2	11000000	380000	10620000
3	80000	56000	24000
9	70000		
11	20000000		
12	100300		
14	1000000		
18	665000		
20		1000000	

9 rows selected.

From this query, we can see that three parcels (2, 3, and 20) in the **mytax** table have qualifying losses from fires. However, since parcel 20 has a NULL value for **bldval**, its building value will remain NULL.

Now, we can attempt to update the **mytax** table:

```
UPDATE mytax t
  SET t.bldval = t.bldval -
      ( SELECT SUM(f.estloss)
        FROM fires f
        WHERE f.parcelid = t.parcelid)
 WHERE EXISTS
      ( SELECT NULL
        FROM fires f, mytax t
        WHERE f.parcelid = t.parcelid);
```

9 rows updated.

Notice that all nine rows in **mytax** were updated, not just the three with qualifying fires. If we look at the **tax** table now, we'll see that many of the building values have been wiped out and set to NULL:

```
SELECT t.parcelid, t.bldval, SUM(f.estloss) losses
FROM mytax t, fires f
WHERE t.parcelid = f.parcelid (+)
GROUP BY t.parcelid, t.bldval
ORDER BY t.parcelid;
```

PARCELID	BLDVAL	LOSSES
1		
2	10620000	380000
3	24000	56000
9		
11		
12		
14		
18		
20		1000000

9 rows selected.

**Question I-4(a) (7 points).** Why were all nine rows in the table updated instead of only the three we wanted (parcels 2, 3, and 20, that had fires)?

**Question I-4(b) (8 points).** Rewrite the UPDATE statement above so that it updates only the three parcels in the **mytax** table that had fires. Run your UPDATE statement, then rerun the last SELECT statement above to show the results from the update. Remember to ROLLBACK any updates that don't work correctly before running the next update. (If you run into too many problems, remember that you can DROP the **mytax** table and CREATE it again.) Show your SQL UPDATE statement and the results from the last query above after the update.

## Part II: Writing SQL Statements (55 points)

For all the questions in this part, show us the SQL and SQL\*Plus statement(s) you used and their results. We cannot give you partial credit if we are unable to follow your work.

**Question II-1(a) (10 points).**

Using the ZONING database, write a query that finds:

- by neighborhood *name*
- the count of cases
- with a known board decision

- that propose to change the land use from 'VACANT LAND' to 'OFFICE'
- for those neighborhoods that have more than one such case.

Use the text descriptions of the land uses, *not* numeric codes. Sort by the counts in descending order.

### Question II-1(b) (15 points).

Now, find the names of all the neighborhoods that have *no* cases with a known board decision that propose to change the land use from 'VACANT LAND' to 'OFFICE', i.e., the neighborhoods *not* returned by the query in Question II-1(a). List the neighborhood names in ascending order.

---

### Question II-2 (15 points).

Using the ZONING database, list the count of cases by neighborhood name where the street name is 'CENTRE'. Include *all* the neighborhoods, whether they have a case or not. List the neighborhood name and the count of cases sorted by the neighborhood name in ascending order.

---

### Question II-3 (15 points).

Consider the SALES table in the 'toy' PARCELS database. Which GRANTORs were also GRANTEE's? Show only their names, in alphabetical order and without duplicates.

---

## Exam Turnin Checklist

- Is your **name** and **Athena ID** at the top of your exam? If not, you'll lose 5 points (2 points if you include one but not the other)!
- Is your completed test saved in a plain text (ASCII) file named **username\_exam.txt** or HTML (web page) file named **username\_exam.html**?
- Did you turn in your test with the **turnin** command? You should use a command like this:

```
turnin -c 11.521 100 username_exam.txt
```

- Did you confirm with an instructor that your exam was received?