SQL Queries of the Zoning Database

The 'toy' parcel database helped us understand basic SQL queries and learn how to join several small tables with at most a few dozen rows. These tables were similar in structure to tables used to record land use and ownership but contained hypothetical data. Then we used the URISA database with real data to get more comfortable with relational thinking and SQL queries that handle one-to-many relationships among a few tables with hundreds of rows. Now, we'll focus on exploring, summarizing, and cross-referencing one larger table (of zoning variance information) that has several dozen columns and several thousand rows. Later, we'll relate this database to even larger tables (the parcel database for Boston) as well as to maps.

The various class web pages on SQL materials describe the zoning database (and associated lookup tables). Here are some of the queries used in class to explore the 'zoning' table data.

The basic steps that we discussed to familiarize ourselves with the data were:

1. Begin with simple SQL selects. Learn some formatting commands to handle columns that are too wide.
2. Use 'group by' to summarize data within groupings and take advantage of SQL's structure and your cut and paste tools to add/modify columns, the 'where' clause and the 'group by's to quickly redo the queries for selected sub-groups.
3. Dump the data into a spreadsheet to allow graphing and summary stats (I did this by spooling SQL output to a file, editing it slightly in emacs, and reading it in using XESS tools to import and parse tables.)
4. Improve the readability of some of the listings by using the lookup tables to substitute neighborhood names, landuses, etc. for various codes.
5. Develop new lookup tables that help categorize various cases using your own reasoning and discoveries. (We'll be doing more of this next week.)

We found it helpful to specify column formats and printing width/pause/... via a 'login.sql' file containing:

```
set linesize 132
set pause on
set pause 'Ready to Rip...'

column applicant format A10
column far format 990
column fname format A10
column streetname format A10

select * from cat;
```

Sample Zoning Database Queries

The next few queries just list selected variables. They'll be useful to cut-and-paste variable names into your own queries and to eyeball the data and experiment with formatting columns:

```
select CASENUMBER,HRINGDATE,APPLICANT,FNAME,STRTNUMBER,
       STRTPREFIX,STREETNAME,STREETTYPE,
       WARD,PRECINCT
  from ZONING;

select CASENUMBER, HRINGDATE, NEIGHBRHD, SUBNGHBRHD,
       CENSUSTRAC,CENSUSBLK,PARCELNUMB,
       EXISTZONIN,ESTCOST,EXISTUSE,PRPSEDUSE,CHNGEOCCUP
  from ZONING;

select CASENUMBER,CHNGEUSE,ADDITION,ERECTSTRUC,ERECTAUXIL,
       LEGALIZE,SUBDIVISIO,PARKING,MISCPURPOS,EXISTUNITS
  from ZONING;

select CASENUMBER,PRPSDUNITS,EXISTSQFT,PRPSDSQFT,
       USEITEM1,UI1,USEITEM2,UI2,USEITEM3,UI3
  from ZONING;

select CASENUMBER, VARIANCE74,
       NONCONF91,NONCONF92,OFFPKG101,LOTSZE141,LOTSZE142,
       LOTSZE143,LOTSZE144,FAR151,HEIGHT161,ROOF168,OSPACE171
  from ZONING;

select CASENUMBER,
       FYARD181,SYARD191,RYARD201,SETBACK211,OFFPKG231,
       OFFLOAD241,IPOD,
       DESIGNREV,DSCOMPLETD,BRARECOM,BOARDDECIS
  from ZONING;
```

The above queries use default widths and column formats. Try this one after running the 'column' formatting commands.

```
select CASENUMBER,HRINGDATE,substr(APPLICANT,1,10) applicant,
       substr(FNAME,1,10) fname, STRTNUMBER,
       STRTPREFIX,substr(STREETNAME,1,10) streetname, STREETTYPE,
       WARD,PRECINCT
  from ZONING;
```

The property description variables are encoded as Yes=2, No=1 and Unknown=-1. Hence, the following queries will compute the percentage of non-missing cases that met each condition:

```
select count(*), 100*avg(CHNGEUSE-1), 100*avg(ADDITION-1),
       100*avg(ERECTSTRUC-1)
  from zoning
 where CHNGEUSE > 0 and ADDITION > 0 and ERECTSTRUC > 0;

select count(*), 100*avg(ERECTAUXIL-1), 100*avg(LEGALIZE-1),
```

```
       100*avg(SUBDIVISIO-1)
  from zoning
 where ERECTAUXIL > 0 and LEGALIZE > 0 and SUBDIVISIO > 0;

select count(*), 100*avg(PARKING-1), 100*avg(MISCPURPOS-1)
  from zoning
 where PARKING > 0 and MISCPURPOS > 0;
```

Most of the possible zoning code violations are encoded similarly. Here is another query to compute percentages for some Yes/No encoded variables: (Note, e.g., that `SYARD181` indicates whether the side yard regulations in chapter 181 are potentially violated.)

```
select count(*), 100*avg(FYARD181-1) pct_fyard181, 100*avg(SYARD191-1),
       100*avg(RYARD201-1), 100*avg(SETBACK211-1)
  from ZONING
 where FYARD181>0 and SYARD191>0 and RYARD201>0
   and SETBACK211>0;
```

Here's the SQL query that could be used (with wide/long linesize/pagesize settings) to dump most of the table into a large, flat ASCII file for input into a spreadsheet.

```
select CASENUMBER,HRINGDATE,APPLICANT,FNAME,STRTNUMBER,
       STRTPREFIX,STREETNAME,STREETTYPE,
       WARD,PRECINCT,
       HRINGDATE, NEIGHBRHD, SUBNGHBRHD,
       CENSUSTRAC,CENSUSBLK,PARCELNUMB,
       EXISTZONIN,ESTCOST,EXISTUSE,PRPSEDUSE,CHNGEOCCUP,
       CHNGEUSE,ADDITION,ERECTSTRUC,ERECTAUXIL,
       LEGALIZE,SUBDIVISIO,PARKING,MISCPURPOS,EXISTUNITS,
       PRPSDUNITS,EXISTSQFT,PRPSDSQFT,
       USEITEM1,UI1,USEITEM2,UI2,USEITEM3,UI3
       VARIANCE74,
       NONCONF91,NONCONF92,OFFPKG101,LOTSZE141,LOTSZE142,
       LOTSZE143,LOTSZE144,FAR151,HEIGHT161,ROOF168,OSPACE171,
       FYARD181,SYARD191,RYARD201,SETBACK211,OFFPKG231,
       OFFLOAD241,IPOD,OTHERVIOLA,
       DESIGNREV,DSCOMPLETD,BRARECOM,BOARDDECIS
  from ZONING;
```

Here's a shorter SQL query that to prepare an ASCII file that could be loaded into a spreadsheet.

```
select CASENUMBER,NONCONF91,NONCONF92,OFFPKG101,
       LOTSZE141,LOTSZE142,FAR151,
       HEIGHT161,ROOF168,OSPACE171,
       SETBACK211,OFFPKG231,IPOD
  from zoning;
```

Here are some useful 'group by' statements:

```
  select existuse, count(*) from zoning
group by existuse;
```

```
  EXISTUSE   COUNT(*)
---------- ----------
        -1         38
         0          2
         1        923
         2         27
         3         59
         4        104
         5         65
         6          1
         7         15
         8        249
         9         38
        10        268
        11         12
```

```
  select prpseduse, count(*) from zoning
group by prpseduse;
```

```
 PRPSEDUSE   COUNT(*)
---------- ----------
        -1         19
         1       1172
         2         35
         3         45
         4        106
         5         48
         6          2
         7         12
         8        307
         9         51
        10          1
        11          3
```

```
  select prpseduse, landuse, count(*)
    from zoning z, use u
   where z.prpseduse = u.use_code
group by prpseduse, landuse;
```

```
 PRPSEDUSE LANDUSE                    COUNT(*)
---------- ------------------------ ----------
        -1 UNKNOWN USE                      19
         1 HOUSING                        1172
         2 OFFICE                           35
         3 RETAIL                           45
         4 COMMERCIAL                      106
         5 EDUCATION/INSTITUTIONAL          48
         6 ENTERTAINMENT                     2
         7 RECREATION                       12
         8 MIXED USE                       307
         9 PARKING                          51
        10 VACANT LAND                       1
        11 OTHER LAND USE                    3
```

Note the above table is identical to the previous one except that the text description of each code is added via the 'lookup' in the USE table.

```
  select existuse,PRPSEDUSE,  count(*) from zoning
   where BRARECOM > 0
group by existuse, PRPSEDUSE
order by existuse, PRPSEDUSE;

  select chngeoccup, count(*) from zoning
group by chngeoccup;

CHNGEOCCUP   COUNT(*)
---------- ----------
        -1          4
         1       1042
         2        754
         8          1

  select designrev, count(*) from zoning
group by designrev;

 DESIGNREV   COUNT(*)
---------- ----------
        -1        248
         0        664
         1        591
         2        298

  select dscompletd, count(*) from zoning
group by dscompletd;

DSCOMPLETD   COUNT(*)
---------- ----------
        -1        249
         0        664
         1        695
         2        193

  select brarecom, boarddecis, count(*) from zoning
   where brarecom > 0 and boarddecis > 0
group by brarecom, boarddecis
order by brarecom, boarddecis;
  BRARECOM BOARDDECIS   COUNT(*)
---------- ---------- ----------
         1          1        116
         1          2        154
         1          4         13
         1          5         14
         2          1         82
         2          2        587
         2          4         35
         2          5         31
         2          6          7
         4          1          9
         4          2         60
         4          4         29
         4          5         16
         4          6          4
         5          1         14
         5          2         92
```

5	4	39
5	5	99
5	6	6
6	1	20
6	2	113
6	4	30
6	5	31
6	6	13
7	7	10
33	1	1
33	2	7
33	4	1
33	33	1