

Correlated Update Statements

First, let's create private copies of the PARCELS, OWNERS, and FIRES tables:

```
DROP TABLE myparcels;
CREATE TABLE myparcels
  AS SELECT * FROM parcels;
```

```
DROP TABLE myowners;
CREATE TABLE myowners
  AS SELECT * FROM owners;
```

```
DROP TABLE myfires;
CREATE TABLE myfires
  AS SELECT * FROM fires;
```

A simple update: Change the address of "VANDELAY INDUSTRIES" to "300 ATLANTIC":

```
-- See the old value
SELECT *
  FROM myowners
 WHERE oname = 'VANDELAY INDUSTRIES';

-- Perform the update
UPDATE myowners
  SET address = '300 ATLANTIC'
 WHERE ONAME = 'VANDELAY INDUSTRIES';

-- Check our work
SELECT *
  FROM myowners
 WHERE oname = 'VANDELAY INDUSTRIES';

-- Commit the change permanently
COMMIT;
```

A more complicated update:

```
-- Look at estimated losses by ZIP code
SELECT f.parcelid, f.fdate, p.zip, f.estloss
  FROM myfires f, myparcels p
 WHERE f.parcelid = p.parcelid
 ORDER BY p.zip, f.parcelid;

-- Increase the losses for parcels in ZIP 02130 by 50%
UPDATE myfires f
  SET estloss = estloss * 1.50
 WHERE EXISTS
   (SELECT NULL
     FROM myparcels p
```

```

        WHERE p.parcelid = f.parcelid
              AND p.zip = '02130');

-- Check the result
SELECT f.parcelid, f.fdate, p.zip, f.estloss
       FROM myfires f, myparcels p
        WHERE f.parcelid = p.parcelid
 ORDER BY p.zip, f.parcelid;

-- Rollback
ROLLBACK;

-- Check again
SELECT f.parcelid, f.fdate, p.zip, f.estloss
       FROM myfires f, myparcels p
        WHERE f.parcelid = p.parcelid
 ORDER BY p.zip, f.parcelid;

```

The most advanced form uses a subquery in the SET clause too.

```

-- Set the losses for parcels in ZIP 02130 to the ZIP code
UPDATE myfires f
   SET estloss =
       (SELECT TO_NUMBER(p.zip)
        FROM myparcels p
         WHERE p.parcelid = f.parcelid)
 WHERE EXISTS
       (SELECT NULL
        FROM myparcels p
         WHERE p.parcelid = f.parcelid
          AND p.zip = '02130');

-- Check the result
SELECT f.parcelid, f.fdate, p.zip, f.estloss
       FROM myfires f, myparcels p
        WHERE f.parcelid = p.parcelid
 ORDER BY p.zip, f.parcelid;

-- Commit the change
COMMIT;

```

Note that it is important to have a WHERE clause in correlated updates to avoid Oracle errors.
