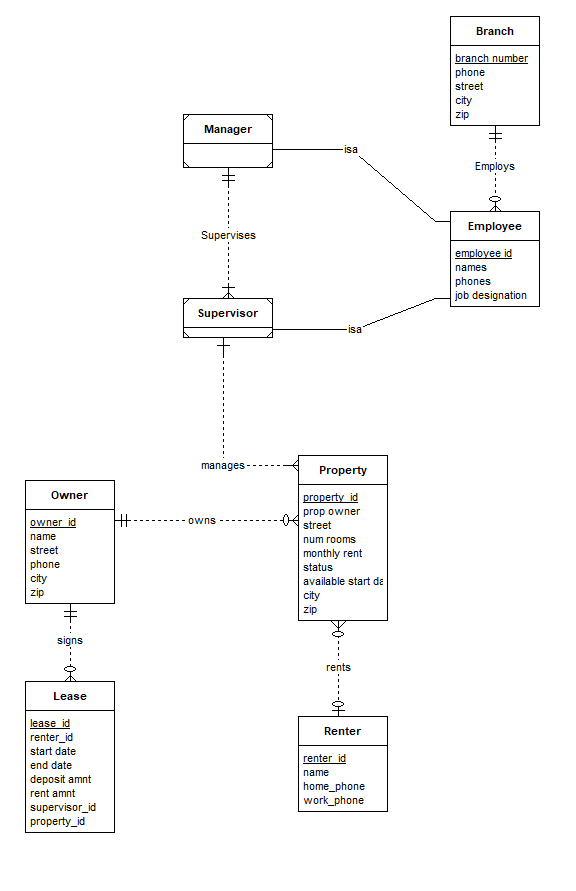
Gabriel Kralik

Happy Renters Final Project



**Functional Dependencies:**

* Branch:
  + branch\_number 🡪phone,street,city,zip
* Employee:
  + employee\_id 🡪 name,phone,job
* Employs:
  + employee\_id 🡪 branch\_number
* Supervises:
  + Supervisor\_id🡪manager\_id
* Manages:
  + prop\_id,supervisor\_id 🡪supervisor\_id
  + prop\_id,supervisor\_id 🡪prop\_id
* Owner:
  + owner\_id 🡪 name,street,city,zip
* Property:
  + prop\_id 🡪 street,city,zip,num\_rooms,monthly\_rent,status,avail\_start\_date
* Owns:
  + Prop\_id🡪owner\_id
* Renter:
  + renter\_id 🡪 name,home\_phone,work\_phone
* Lease:
  + lease\_id 🡪lease\_id 🡪 renter\_id, start\_date, end\_date, deposit\_amt, supervisor\_id, prop\_id

Resulting Tables: (Underline = primary key, FK = foreign key)

**Branch** (branch\_number, phone, street, city, zip)

**Employee** (employee\_id, name, phone, job)

**Employs** (branch\_number (FK), employee\_id (FK))

**Supervises** (manager\_id (FK), supervisor\_id (FK))

**Owner** (owner\_id, name, street, city, zip, phone)

**Property** (prop\_id, street, city, zip, num\_rooms, monthly\_rent, status, avail\_start\_date)

**Owns** (prop\_id (FK), owner\_id (FK))

**Manages** (supervisor\_id (FK), prop\_id (FK))

**Renter** (renter\_id, rname, rhome, rwork)

**Lease** (lease\_id, renter\_id (FK), prop\_id (FK), supervisor\_id (FK), start\_date, end\_date, deposit\_amount)

To easily perform certain queries and store less data for certain relationship tables (owns and lease), I created special keys for leases, owners, and renters. That way, I would not have to create any unnecessary assumptions. As for restrictions, there must only be one manager per branch, this restriction is imposed by a trigger that checks whenever an insertion is made to the employs table. We check that a lease agreement is always between six months and one year inclusive by keeping a check constraint built into the table’s CREATE statement. In the procedure to create a lease agreement, there is code to change the property’s status to *leased* when an agreement is successfully created, and the same procedure updates the rent of the house for the next lease. Rather than create a *rents* table and a separate table for owners who have signed rents, the lease table functions as a sort of super table containing all that information and preventing us from creating redundant data without losing any important information.

Demo output:

SQL> start execs.sql

Property ID: 0 Address: Towne Valley Drive Santa Clara 95053 Monthly Rent:

$1412.34 Number of Rooms: 2 Owner Name: George Washington

Property ID: 2 Address: Desert Hill Road Santa Clara 95050 Monthly Rent: $1234.5

Number of Rooms: 2 Owner Name: Thomas Jefferson

Property ID: 4 Address: Oak Pine Way Santa Clara 95053 Monthly Rent: $2100

Number of Rooms: 3 Owner Name: James Monroe

Property ID: 6 Address: Parrot Street Santa Clara 95049 Monthly Rent: $3300

Number of Rooms: 3 Owner Name: George Washington

Manager: John Smith

PL/SQL procedure successfully completed.

Property ID: 3 Address: Blue Ridge Parkway Los Angeles 90003 Monthly Rent: $9000

Number of Rooms: 7 Owner Name: James Madison

Property ID: 5 Address: Dead End Path Los Angeles 90009 Monthly Rent: $1300

Number of Rooms: 1 Owner Name: John Q Adams

Property ID: 7 Address: Danny Boy Lane Los Angeles 90010 Monthly Rent: $2900

Number of Rooms: 3 Owner Name: John Adams

Manager: Jane Baker

PL/SQL procedure successfully completed.

Supervisor: Jane Doe

Property ID: 0 Address: Towne Valley Drive Santa Clara

95053 Monthly Rent: $1412.34 Number of Rooms: 2 Owner: George Washington

Supervisor: Jack Mansfield

Property ID: 1 Address: Liberty Road Los Angeles

90005 Monthly Rent: $1006.71 Number of Rooms: 4 Owner: John Adams

Supervisor: Jane Doe

Property ID: 2 Address: Desert Hill Road Santa Clara 95050

Monthly Rent: $1234.5 Number of Rooms: 2 Owner: Thomas Jefferson

Supervisor: Jack Mansfield

Property ID: 3 Address: Blue Ridge Parkway Los

Angeles 90003 Monthly Rent: $9000 Number of Rooms: 7 Owner: James Madison

Supervisor: Jane Doe

Property ID: 4 Address: Oak Pine Way Santa Clara 95053

Monthly Rent: $2100 Number of Rooms: 3 Owner: James Monroe

Supervisor: Jeff Donohue

Property ID: 5 Address: Dead End Path Los Angeles 90009

Monthly Rent: $1300 Number of Rooms: 1 Owner: John Q Adams

Supervisor: John Deer

Property ID: 6 Address: Parrot Street Santa Clara 95049

Monthly Rent: $3300 Number of Rooms: 3 Owner: George Washington

Supervisor: Jeff Donohue

Property ID: 7 Address: Danny Boy Lane Los Angeles

90010 Monthly Rent: $2900 Number of Rooms: 3 Owner: John Adams

PL/SQL procedure successfully completed.

Property ID: 0 Address: Towne Valley Drive Santa Clara 95053 Monthly Rent:

$1412.34 Number of Rooms: 2 Owner: George Washington

Property ID: 6 Address: Parrot Street Santa Clara 95049 Monthly Rent: $3300

Number of Rooms: 3 Owner: George Washington

PL/SQL procedure successfully completed.

Property ID: 0 Address: Towne Valley Drive Santa Clara 95053 Monthly Rent:

$1412.34 Number of Rooms: 2 Owner Name: George Washington

Property ID: 2 Address: Desert Hill Road Santa Clara 95050 Monthly Rent: $1234.5

Number of Rooms: 2 Owner Name: Thomas Jefferson

PL/SQL procedure successfully completed.

Property ID: 3 Address: Blue Ridge Parkway Los Angeles 90003 Monthly Rent: $9000

Number of Rooms: 7 Owner Name: James Madison

Property ID: 7 Address: Danny Boy Lane Los Angeles 90010 Monthly Rent: $2900

Number of Rooms: 3 Owner Name: John Adams

PL/SQL procedure successfully completed.

Number of available houses: 7

PL/SQL procedure successfully completed.

COUNT(\*)

----------

7

MONTHLY\_RENT

------------

1412.34

PL/SQL procedure successfully completed.

COUNT(\*)

----------

6

MONTHLY\_RENT

------------

1553.57

Renter Name: Dom Pedro I

Renter home phone: 2390000014

Renter work phone:

2390000015

Starting Date of Lease: 25-DEC-17

Ending Date of Lease:

23-JUL-18

Deposit Amount: $1776.71

Rent Amount: $1776.71 per month

Supervisor:

John Deer

Renter Name: Dom Pedro I

Renter home phone: 2390000014

Renter work phone:

2390000015

Starting Date of Lease: 08-JUN-18

Ending Date of Lease:

08-FEB-19

Deposit Amount: $1412.34

Rent Amount: $1412.34 per month

Supervisor:

Jane Doe

PL/SQL procedure successfully completed.

Renter: Dom Pedro I

PL/SQL procedure successfully completed.

Average Rent among all properties in santa clara: $2047.02

Average Rent of available properties: $2211.5

Average Rent of leased properties: $1553.57

PL/SQL procedure successfully completed.

Property ID: 1 Address: Liberty Road Los Angeles 90005 Monthly Rent: $1006.71

Number of Rooms: 4 Owner: John Adams Start Date: 23-JUL-18

PL/SQL procedure successfully completed.

SQL> spool off;

**SQL AND PL/SQL Code:**

To run and see the system in action, run setup.sql then execs.sql.

makeTables.sql:

CREATE TABLE Branch (

branch\_number CHAR(8) PRIMARY KEY,

phone CHAR(10),

street VARCHAR(32),

city VARCHAR(32),

zip CHAR(5)

);

CREATE TABLE Employee (

employee\_id CHAR(8) PRIMARY KEY,

name VARCHAR(32),

phone CHAR(10),

job VARCHAR(11),

CHECK (job in ('manager','supervisor'))

);

CREATE TABLE Owner (

owner\_id CHAR(8) PRIMARY KEY,

name VARCHAR(32),

street VARCHAR(32),

city VARCHAR(32),

zip CHAR(5),

phone CHAR(10)

);

CREATE TABLE Property (

prop\_id Integer PRIMARY KEY,

street VARCHAR(32),

city VARCHAR(32),

zip CHAR(5),

num\_rooms Integer,

monthly\_rent NUMBER(\*,2),

status VARCHAR(11) DEFAULT 'available',

avail\_start\_date DATE,

CHECK (status in ('available','leased'))

);

CREATE TABLE Employs (

branch\_number CHAR(8),

employee\_id CHAR(8),

FOREIGN KEY (branch\_number) REFERENCES Branch(branch\_number),

FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id)

);

CREATE TABLE Supervises (

manager\_id CHAR(8),

supervisor\_id CHAR(8) PRIMARY KEY,

FOREIGN KEY (manager\_id) REFERENCES Employee(employee\_id),

FOREIGN KEY (supervisor\_id) REFERENCES Employee(employee\_id)

);

CREATE TABLE Manages (

supervisor\_id CHAR(8),

prop\_id Integer PRIMARY KEY,

FOREIGN KEY (supervisor\_id) REFERENCES Employee(employee\_id) ON DELETE CASCADE,

FOREIGN KEY (prop\_id) REFERENCES Property(prop\_id) ON DELETE CASCADE

);

CREATE TABLE Owns (

prop\_id Integer PRIMARY KEY,

owner\_id CHAR(8),

FOREIGN KEY (prop\_id) REFERENCES Property(prop\_id) ON DELETE CASCADE,

FOREIGN KEY (owner\_id) REFERENCES Owner(owner\_id)

);

CREATE TABLE Renter (

rname VARCHAR(32),

rhome CHAR(10),

rwork CHAR(10),

renter\_id CHAR(8) PRIMARY KEY

);

CREATE TABLE lease (

renter\_id CHAR(8),

start\_date DATE,

end\_date DATE,

deposit\_amt NUMBER(\*,2),

supervisor\_id CHAR(8),

lease\_id Integer PRIMARY KEY,

prop\_id Integer,

CHECK ((end\_date - start\_date <= 365) AND (end\_date - start\_date >= 182)),

FOREIGN KEY (prop\_id) REFERENCES Property(prop\_id) ON DELETE CASCADE,

FOREIGN KEY (renter\_id) REFERENCES renter(renter\_id),

FOREIGN KEY (supervisor\_id) REFERENCES Employee(employee\_id)

);

Inserts.sql:

insert into branch values ('10000001','2390000000','El Camino Real','Santa Clara','95053');

insert into branch values ('10000002','2390000001','Baker Street','Los Angeles','90003');

insert into employee values ('00000001','John Smith','2390000002','manager');

insert into employee values ('00000002','Jane Doe','2390000003','supervisor');

insert into employee values ('00000003','John Deer','2390000004','supervisor');

insert into employee values ('00000004','Jane Baker','2390000005','manager');

insert into employee values ('00000005','Jack Mansfield','2390000006','supervisor');

insert into employee values ('00000006','Jeff Donohue','2390000007','supervisor');

insert into employs values ('10000001','00000001');

insert into employs values ('10000001','00000002');

insert into employs values ('10000001','00000003');

insert into employs values ('10000002','00000004');

insert into employs values ('10000002','00000005');

insert into employs values ('10000002','00000006');

insert into supervises values ('00000001','00000002');

insert into supervises values ('00000001','00000003');

insert into supervises values ('00000004','00000005');

insert into supervises values ('00000004','00000006');

insert into owner values ('20000001','George Washington','Virgina Avenue','Norfolk','12340','2390000008');

insert into owner values ('20000002','John Adams','Massachusetts Road','Boston','12341','2390000009');

insert into owner values ('20000003','Thomas Jefferson','Delaware Street','Dover','12342','2390000010');

insert into owner values ('20000004','James Madison','New Hampshire Road','Concord','12343','2390000011');

insert into owner values ('20000005','James Monroe','New York Way','Buffalo','12344','2390000012');

insert into owner values ('20000006','John Q Adams','Vermont','Montpelier','12345','2390000013');

insert into property values (0,'Towne Valley Drive','Santa Clara','95053',2,1412.34,'available',

TO\_DATE('01-JUN-18','DD-MON-YY'));

insert into property values (1,'Liberty Road','Los Angeles','90005',4,1006.71,'leased',

TO\_DATE('23-JUL-18','DD-MON-YY'));

insert into property values (2,'Desert Hill Road','Santa Clara','95050',2,1234.50,'available',

TO\_DATE('01-JUN-18','DD-MON-YY'));

insert into property values (3,'Blue Ridge Parkway','Los Angeles','90003',7,9000.00,'available',

TO\_DATE('01-JUN-18','DD-MON-YY'));

insert into property values (4,'Oak Pine Way','Santa Clara','95053',3,2100.00,'available',

TO\_DATE('01-JUN-18','DD-MON-YY'));

insert into property values (5,'Dead End Path','Los Angeles','90009',1,1300.00,'available',

TO\_DATE('01-JUN-18','DD-MON-YY'));

insert into property values (6,'Parrot Street','Santa Clara','95049',3,3300.00,'available',

TO\_DATE('01-JUN-18','DD-MON-YY'));

insert into property values (7,'Danny Boy Lane','Los Angeles','90010',3,2900.00,'available',

TO\_DATE('01-JUN-18','DD-MON-YY'));

insert into manages values ('00000002',0);

insert into manages values ('00000002',2);

insert into manages values ('00000002',4);

insert into manages values ('00000003',6);

insert into manages values ('00000005',1);

insert into manages values ('00000005',3);

insert into manages values ('00000006',5);

insert into manages values ('00000006',7);

insert into owns values (0,'20000001');

insert into owns values (1,'20000002');

insert into owns values (2,'20000003');

insert into owns values (3,'20000004');

insert into owns values (4,'20000005');

insert into owns values (5,'20000006');

insert into owns values (6,'20000001');

insert into owns values (7,'20000002');

insert into renter values ('Dom Pedro I','2390000014','2390000015','30000001');

insert into renter values ('Getulio Vargas','2390000016','2390000017','30000002');

insert into renter values ('Michel Temer','2390000018','2390000019','30000003');

insert into renter values ('Dilma Rousseff','2390000020','2390000021','30000004');

insert into renter values ('Luiz Inacio Lula da Silva','2390000022','2390000023','30000005');

insert into lease values ('30000001',TO\_DATE('25-DEC-2017','DD-MON-YYYY'),TO\_DATE('23-JUL-2018','DD-MON-YYYY'),

1006.71,'00000003',0,1);

manager\_constraint.sql:

CREATE OR REPLACE Trigger manager\_constraint

BEFORE INSERT on Employs

FOR EACH ROW

DECLARE

l\_new\_job employee.job%type;

l\_man\_count Integer := 0;

BEGIN

SELECT COUNT(\*) INTO l\_man\_count FROM employs, employee

where employs.employee\_id = employee.employee\_id AND job='manager'

AND employs.branch\_number = :new.branch\_number;

SELECT job INTO l\_new\_job FROM employee WHERE employee\_id=:new.employee\_id;

IF (l\_man\_count > 0 AND l\_new\_job='manager') THEN

RAISE\_APPLICATION\_ERROR(-20000, 'May only have one manager per branch');

ELSE

DBMS\_OUTPUT.put\_line('Manager added to selected branch');

END IF;

END;

/

SHOW ERRORS

t1.sql:

CREATE OR REPLACE Procedure t1(branch\_num in VARCHAR)

As

manager\_name employee.name%type;

Cursor prop\_cur IS

SELECT prop\_id,street,city,zip,monthly\_rent,num\_rooms FROM property WHERE status='available' AND prop\_id IN

(SELECT prop\_id FROM manages WHERE supervisor\_id IN

(SELECT employee\_id FROM employs WHERE branch\_number=branch\_num) );

l\_propid property.prop\_id%type;

l\_prop\_owner owner.name%type;

l\_numrooms property.num\_rooms%type;

l\_street property.street%type;

l\_city property.city%type;

l\_zip property.zip%type;

l\_rent property.monthly\_rent%type;

BEGIN

OPEN prop\_cur;

loop

FETCH prop\_cur into l\_propid,l\_street,l\_city,l\_zip,l\_rent,l\_numrooms;

EXIT WHEN prop\_cur%notfound;

SELECT name INTO l\_prop\_owner FROM owner WHERE owner\_id =

(SELECT owner\_id FROM owns WHERE prop\_id = l\_propid) ;

DBMS\_OUTPUT.put\_line('Property ID: ' || l\_propid || ' Address: ' ||

l\_street || ' ' || l\_city || ' ' || l\_zip || ' Monthly Rent: $' || l\_rent || ' Number of Rooms: '

|| l\_numrooms || ' Owner Name: ' || l\_prop\_owner || chr(10) );

END LOOP;

CLOSE prop\_cur;

SELECT name INTO manager\_name from employee WHERE job = 'manager' AND employee\_id IN

(SELECT employee\_id FROM employs WHERE branch\_number = branch\_num);

DBMS\_OUTPUT.put\_line('Manager: ' || manager\_name);

END;

/

t2.sql:

CREATE OR REPLACE Procedure t2

AS

l\_supervisor\_name employee.name%type;

Cursor prop\_cur IS

SELECT prop\_id,street,city,zip,monthly\_rent,num\_rooms FROM property;

l\_propid property.prop\_id%type;

l\_street property.street%type;

l\_city property.city%type;

l\_zip property.zip%type;

l\_rent property.monthly\_rent%type;

l\_numrooms property.num\_rooms%type;

l\_prop\_owner owner.name%type;

BEGIN

OPEN prop\_cur;

LOOP

FETCH prop\_cur into l\_propid,l\_street,l\_city,l\_zip,l\_rent,l\_numrooms;

EXIT WHEN prop\_cur%notfound;

SELECT name INTO l\_supervisor\_name FROM employee WHERE employee\_id =

(SELECT supervisor\_id FROM manages WHERE prop\_id = l\_propid);

SELECT name INTO l\_prop\_owner FROM owner WHERE owner\_id =

(SELECT owner\_id FROM owns WHERE prop\_id = l\_propid);

DBMS\_OUTPUT.put\_line('Supervisor: ' || l\_supervisor\_name || chr(10) || 'Property ID: '

|| l\_propid || ' Address: ' || l\_street || ' ' || l\_city || ' ' || l\_zip ||

' Monthly Rent: $' || l\_rent || ' Number of Rooms: ' || l\_numrooms || ' Owner: '

|| l\_prop\_owner || chr(10));

END LOOP;

CLOSE prop\_cur;

END;

/

Show Errors

t3.sql:

CREATE OR REPLACE PROCEDURE t3(in\_owner\_id in VARCHAR, in\_branchid in VARCHAR)

As

Cursor prop\_cur IS

SELECT prop\_id,street,city,zip,monthly\_rent,num\_rooms FROM property WHERE

prop\_id in (SELECT prop\_id FROM owns WHERE owner\_id=in\_owner\_id);

l\_propid property.prop\_id%type;

l\_street property.street%type;

l\_city property.city%type;

l\_zip property.zip%type;

l\_rent property.monthly\_rent%type;

l\_numrooms property.num\_rooms%type;

l\_prop\_owner owner.name%type;

BEGIN

SELECT name INTO l\_prop\_owner FROM owner WHERE owner\_id=in\_owner\_id;

OPEN prop\_cur;

LOOP

FETCH prop\_cur into l\_propid,l\_street,l\_city,l\_zip,l\_rent,l\_numrooms;

EXIT WHEN prop\_cur%notfound;

DBMS\_OUTPUT.put\_line('Property ID: ' || l\_propid || ' Address: ' || l\_street ||

' ' || l\_city || ' ' || l\_zip || ' Monthly Rent: $' || l\_rent || ' Number of Rooms: '

|| l\_numrooms || ' Owner: ' || l\_prop\_owner || chr(10));

END LOOP;

CLOSE prop\_cur;

END;

/

Show Errors;

t4.sql:

CREATE OR REPLACE Procedure t4(in\_city in VARCHAR, in\_numrooms in Integer, in\_lo\_rent in number,

in\_hi\_rent in number)

As

Cursor prop\_cur IS

SELECT prop\_id,street,city,zip,monthly\_rent,num\_rooms FROM property

WHERE upper(city)=upper(in\_city) AND num\_rooms >= in\_numrooms AND status='available';

Cursor prop\_cur\_range IS

SELECT prop\_id,street,city,zip,monthly\_rent,num\_rooms FROM property

WHERE upper(city)=upper(in\_city) AND num\_rooms >= in\_numrooms AND status='available'

AND monthly\_rent <= in\_hi\_rent AND monthly\_rent >= in\_lo\_rent;

l\_prop\_owner owner.name%type;

l\_propid property.prop\_id%type;

l\_street property.street%type;

l\_city property.city%type;

l\_zip property.zip%type;

l\_rent property.monthly\_rent%type;

l\_numrooms property.num\_rooms%type;

BEGIN

IF (in\_hi\_rent = 0) THEN

OPEN prop\_cur;

LOOP

FETCH prop\_cur into l\_propid,l\_street,l\_city,l\_zip,l\_rent,l\_numrooms;

EXIT WHEN prop\_cur%notfound;

SELECT name INTO l\_prop\_owner FROM owner WHERE owner\_id =

(SELECT owner\_id FROM owns WHERE prop\_id = l\_propid);

DBMS\_OUTPUT.put\_line('Property ID: ' || l\_propid || ' Address: ' ||

l\_street || ' ' || l\_city || ' ' || l\_zip || ' Monthly Rent: $' || l\_rent ||

' Number of Rooms: ' || l\_numrooms || ' Owner Name: ' || l\_prop\_owner || chr(10) );

END LOOP;

CLOSE prop\_cur;

ELSE

OPEN prop\_cur\_range;

LOOP

FETCH prop\_cur\_range into l\_propid,l\_street,l\_city,l\_zip,l\_rent,l\_numrooms;

EXIT WHEN prop\_cur\_range%notfound;

SELECT name INTO l\_prop\_owner FROM owner WHERE owner\_id =

(SELECT owner\_id FROM owns WHERE prop\_id = l\_propid);

DBMS\_OUTPUT.put\_line('Property ID: ' || l\_propid || ' Address: ' ||

l\_street || ' ' || l\_city || ' ' || l\_zip || ' Monthly Rent: $' || l\_rent ||

' Number of Rooms: ' || l\_numrooms || ' Owner Name: ' || l\_prop\_owner || chr(10));

END LOOP;

CLOSE prop\_cur\_range;

END IF;

END;

/

Show Errors;

T5.sql:

CREATE OR REPLACE Procedure t5

As

l\_count Integer := 0;

BEGIN

SELECT count(\*) into l\_count FROM property WHERE status='available';

DBMS\_OUTPUT.put\_line('Number of available houses: ' || l\_count);

END;

/

Show Errors;

t6.sql:

CREATE OR REPLACE Procedure t6(in\_renter\_id IN VARCHAR, in\_prop\_id IN Integer, in\_start IN DATE,

in\_end IN DATE)

As

l\_rname renter.rname%type;

l\_rhome renter.rhome%type;

l\_rwork renter.rwork%type;

l\_status property.status%type;

l\_rent property.monthly\_rent%type;

l\_supid manages.supervisor\_id%type;

l\_supname employee.name%type;

l\_leaseid lease.lease\_id%type;

BEGIN

SELECT status INTO l\_status FROM property WHERE prop\_id = in\_prop\_id;

IF (l\_status = 'available') THEN

SELECT rname,rhome,rwork into l\_rname,l\_rhome,l\_rwork FROM renter where renter\_id=in\_renter\_id;

SELECT monthly\_rent into l\_rent FROM property where prop\_id=in\_prop\_id;

SELECT supervisor\_id into l\_supid FROM manages WHERE prop\_id=in\_prop\_id;

SELECT name into l\_supname FROM employee where employee\_id=l\_supid;

SELECT count(\*) INTO l\_leaseid FROM lease;

l\_leaseid :=l\_leaseid + 1;

INSERT INTO lease values (in\_renter\_id,in\_start,in\_end,l\_rent,l\_supid,l\_leaseid,in\_prop\_id);

UPDATE property

set status='leased', monthly\_rent=1.1\*l\_rent, avail\_start\_date=in\_end

WHERE prop\_id=in\_prop\_id;

ELSE

RAISE\_APPLICATION\_ERROR(-20000, 'Property not available!');

END IF;

END;

/

SHOW ERRORS

t7.sql:

CREATE OR REPLACE Procedure t7(in\_renterid IN VARCHAR)

As

Cursor lease\_cur IS

SELECT start\_date,end\_date,deposit\_amt,supervisor\_id,lease\_id,prop\_id FROM

lease WHERE renter\_id=in\_renterid;

l\_start lease.start\_date%type;

l\_end lease.end\_date%type;

l\_deposit lease.deposit\_amt%type;

l\_supid lease.supervisor\_id%type;

l\_leaseid lease.lease\_id%type;

l\_propid lease.prop\_id%type;

l\_renter\_name renter.rname%type;

l\_supname employee.name%type;

l\_rhome renter.rhome%type;

l\_rwork renter.rwork%type;

BEGIN

SELECT rname INTO l\_renter\_name FROM renter WHERE renter\_id = in\_renterid;

OPEN lease\_cur;

LOOP

FETCH lease\_cur into l\_start,l\_end,l\_deposit,l\_supid,l\_leaseid,l\_propid;

EXIT WHEN lease\_cur%notfound;

SELECT rhome into l\_rhome FROM renter where renter\_id=in\_renterid;

SELECT rwork into l\_rwork FROM renter where renter\_id=in\_renterid;

SELECT name into l\_supname FROM employee where employee\_id=l\_supid;

DBMS\_OUTPUT.put\_line('Renter Name: ' || l\_renter\_name || chr(10)

|| 'Renter home phone: ' || l\_rhome || chr(10)

|| 'Renter work phone: ' || l\_rwork || chr(10)

|| 'Starting Date of Lease: ' || l\_start || chr(10)

|| 'Ending Date of Lease: ' || l\_end || chr(10)

|| 'Deposit Amount: $' || l\_deposit || chr(10)

|| 'Rent Amount: $' || l\_deposit || ' per month' || chr(10)

|| 'Supervisor: ' || l\_supname || chr(10) );

END LOOP;

CLOSE lease\_cur;

END;

/

Show Errors;

t8.sql:

CREATE OR REPLACE Procedure t8

As

Cursor renter\_cur IS

SELECT renter\_id FROM lease group by renter\_id HAVING count(\*) > 1;

l\_renterid lease.renter\_id%type;

l\_renter\_name renter.rname%type;

BEGIN

OPEN renter\_cur;

LOOP

FETCH renter\_cur into l\_renterid;

EXIT WHEN renter\_cur%notfound;

SELECT rname into l\_renter\_name FROM renter WHERE renter\_id=l\_renterid;

DBMS\_OUTPUT.put\_line('Renter: ' || l\_renter\_name || chr(10));

END LOOP;

CLOSE renter\_cur;

END;

/

Show Errors;

t9.sql:

CREATE OR REPLACE PROCEDURE t9(in\_city in VARCHAR)

As

l\_total\_avg property.monthly\_rent%type;

l\_avail\_avg property.monthly\_rent%type;

l\_leased\_avg property.monthly\_rent%type;

BEGIN

SELECT AVG(monthly\_rent) INTO l\_total\_avg FROM property WHERE UPPER(city)=UPPER(in\_city);

SELECT AVG(monthly\_rent) INTO l\_avail\_avg FROM property WHERE UPPER(city)=UPPER(in\_city)

AND status='available';

SELECT AVG(monthly\_rent) INTO l\_leased\_avg FROM property WHERE UPPER(city)=UPPER(in\_city)

AND status='leased';

DBMS\_OUTPUT.put\_line('Average Rent among all properties in ' || in\_city || ': $'

|| l\_total\_avg || chr(10));

DBMS\_OUTPUT.put\_line('Average Rent of available properties: $' || l\_avail\_avg || chr(10));

DBMS\_OUTPUT.put\_line('Average Rent of leased properties: $' || l\_leased\_avg || chr(10));

END;

/

Show Errors

t10.sql:

CREATE OR REPLACE Procedure t10

As

cursor prop\_cur IS

SELECT prop\_id,street,city,zip,monthly\_rent,num\_rooms,avail\_start\_date FROM property WHERE

ABS(SYSDATE - avail\_start\_date) <= 60 AND status='leased';

l\_propid property.prop\_id%type;

l\_street property.street%type;

l\_city property.city%type;

l\_zip property.zip%type;

l\_rent property.monthly\_rent%type;

l\_numrooms property.num\_rooms%type;

l\_owner owner.name%type;

l\_date property.avail\_start\_date%type;

BEGIN

OPEN prop\_cur;

LOOP

FETCH prop\_cur into l\_propid,l\_street,l\_city,l\_zip,l\_rent,l\_numrooms,l\_date;

EXIT WHEN prop\_cur%notfound;

SELECT name INTO l\_owner FROM owner WHERE owner\_id =

(SELECT owner\_id FROM owns WHERE prop\_id = l\_propid);

DBMS\_OUTPUT.put\_line('Property ID: ' || l\_propid || ' Address: ' || l\_street ||

' ' || l\_city || ' ' || l\_zip || ' Monthly Rent: $' || l\_rent || ' Number of Rooms: '

|| l\_numrooms || ' Owner: ' || l\_owner || ' Start Date: ' || l\_date || chr(10));

END LOOP;

CLOSE prop\_cur;

END;

/

Show Errors