

Project Report

This document is broken into sections based on phases as detailed by the Project description. The graphs from Phase B were not used in final implementation, and the graphs used are included in one of the appendices.

Phase A

Mars Reality, Inc requires a web or java application interface to a backend database. This database application needs to be lightweight and quick as all employees throughout the company will be utilizing the same information, often simultaneously. This database is required to hold information on the company's branches, employees, clients, properties, property owners, leases, advertising, and property viewings. The application needs to be able to query and report on aspects concerning employees in a branch, employees in the company, client data, and property data to include viewings, rents, locations, as well as property owners. This reality database application needs to allow employees to update and delete records. Reality database application also needs to be able to automatically update the rent based on Adaptive Rent Services. Reality database application needs to be able to sort the results on the data based on various criteria. Each item in the database will have a unique key.

All the data requirements will need to be modeled in an entity-relationship diagram. This diagram will demonstrate the attributes of the various entities, as well as their relationships such as, client(s) lease a property. There are many intertwined and complicated relationships in the data requirements, as well as a lot of duplicated data. Duplicated data in this context refers to multiple entities sharing attributes. There is a hierarchy relationship in property owners, whereby the owner can be private or a business. There are many specific data relationship requirements regarding the quantity of entities which can be related to another entity such as, an associate may be assigned a maximum of 30 properties. These relationship restrictions will need to be accounted for as rules in the database, which could trigger warnings if an employee violates them. The employee entity may or may not have a supervisor, and a supervisor is an employee. Each branch also has a manager, which is a supervisor, and also an employee. Mapping this relationship accurately may be tricky.

The reports will have to be formatted differently depending on the fields in the generated results table. All reports will have to be built in a standard printout fashion which is clearly legible.

Phase B

Graph's A & B depict the relationships between Employees, Clients, Property Owners, Branches, Newspaper Advertisements, Property, and Leases. The following are

constraints on the relationships: each employee can manage at most 30 properties; each lease has a minimum duration of 3 months with a maximum of 24. A client entity can only have one view relationship of a property in the same day. There are no other restraints on the system beyond those depicted in the graphs. Graphs A and B are in Appendix A.

Phase C

Relations for the data modeled for the Mars Reality Database Application are:

Newspaper(newspaper_name, tele, addr, fax, contact)
Branch(branch_num, manager, addr.str, addr.city, addr.st, addr.zip, fax, tele, monthly_bonus, manager_start)
Client(client_num, name, preference, max_rent, employee_name, date_joined, branch_joined)
ClientHomeTele(client_num, home_tele)
ClientWorkTele(client_num, work_tele)
Employee(emp_num, position, branch_num, supervisor, name, salary, addr, sex, dob)
PropOwner(name, private, addr, contact, type)
PropOwnerTele(name, tele)
PropOwnerFax(name, fax)
Property(property_num, owner, rent, fee, bedrooms, bathrooms, sq_footage, type, addr.str, addr.city, addr.st, addr.zip)
Lease(lease_num, property_num, addr, pay_method, deposit, startdate, enddate, rent)
LeaseClientInfo(lease_num, client_name, client_addr)
advertises(property_num, newspaper_name, the_date, cost)
views(client_num, property_num, the_date, comments, client_addr)
obtains(lease_num, client_num)
works_on(lease_num, emp_num)
rents(name, property_num)
works_for(assoc_num, super_num)
works_at(emp_num, branch_num)
manages(emp_num, branch_num)
refers_to(lease_num, property_num)

Of those relations the following were found to be redundant and were removed from the implementation of the database: works_for; works_at; manages; refers_to. These were found redundant because the Employee relation stores the supervisor, the Employee relation also stores the branch_num, the Branch stores the name of the manager, and the Lease entity relation stores the property_num making refers_to superfluous. All relations maintained that the primary key was a super key and could infer all other attributes in the relation. All relations satisfied the requirements of BCNF.

Phase D

The script used to build the SQL tables is rather lengthy and is included in Appendix C. The constraint that an employee can at most work on 30 Leases was never addressed and is only a note in the design. The script for inserting the sample data used for the application is Appendix D. This sample data contains: at least 5 branches in three different states, at least 3 employees, 5 properties, 2 owners, 4 clients, 2 leases, 1 advertisement, and 1 property viewing. This was a substantial effort to generate all the data to be used in testing. An additional script which will delete all rows from all tables and then drop all tables minding referential integrity constraints has also been included as Appendix E. The user interface designed is a basic looping menu driven system, whereby users navigate by selecting corresponding numbers from a list. Another constraint left out of this design was the triggers for the Adaptive Rents Service.

Phase E

The Mars Reality Database Application was written in Java and utilized JDBC to interface with the UMBC GL Oracle 8.1.5 Database Server. The account used by this program belongs to tri1@umbc.edu. There is one query not implemented, Query number 17. To view a baseline of all queries ran with sample data, see Appendix F. A useful index created for the table Property in reference to its type is a good idea because this could group the Properties of the same type together. A frequent query is a search for a certain type of property. A general note that I never created the triggers; the formatting on the reports is rather elementary—although the columns are appropriately labeled. The GL system does not handle cascading, therefore when a delete is entered only that entity and those entities within one degree of separation are removed. Also due to this, users cannot update a primary key.

Phase F

User Manual for Mars Reality Database Application

Run: java MarsDBApp

Requires: CLASSPATH for oracle815.jar driver

Administrator:

Run build.sql script to build the database tables if needed.

All Users:

Enter a username and password to enter the applications' main menu.

Choose:

- 1 – Enter New Data
- 2 – Update Data
- 3 – Delete Data
- 4 – Generate Reports
- 5 – Exit

Enter New Data

A new menu will come up with the following options:

- 1 - Enter a new Branch
- 2 - Enter a new Employee
- 3 - Enter a new Client
- 4 - Enter a new Property Owner
- 5 - Enter a new Property
- 6 - Enter a new Lease
- 7 - Enter a new Advertisement
- 8 - Enter a new Viewing
- 9 - Back

Each of these options, except for #9, will take you through a set of questions which will then generate the new Branch, Employee, etc. The user must generate a unique identification number as an attribute for a Branch, Employee, Client, and Lease.

Update Data

A new menu will come up with the following options:

- 1 - Branch Details
- 2 - Employee Details
- 3 - Lease Details
- 4 - Advertisement Details
- 5 - Owner Details
- 6 - Property Details
- 7 - Client Details
- 8 - Back

Each of these options will require the user to already know the identifying attribute for the object the user would like to update, such as the branch number. With this data the application will return the current information stored and then prompt the user to reenter the data—the user can copy and paste data that hasn't changed instead of retyping everything.

Delete Data

A new menu will come up with the following options:

- 1 - Branch
- 2 - Employee
- 3 - Lease
- 4 - Advertisement
- 5 - Owner
- 6 - Property

- 7 - Client
- 8 - Back

The user will need to know the unique identifying attribute of the item the user wishes removed from the database. With the removal of this object any references to this object within one degree of separation are also removed.

Generate Reports

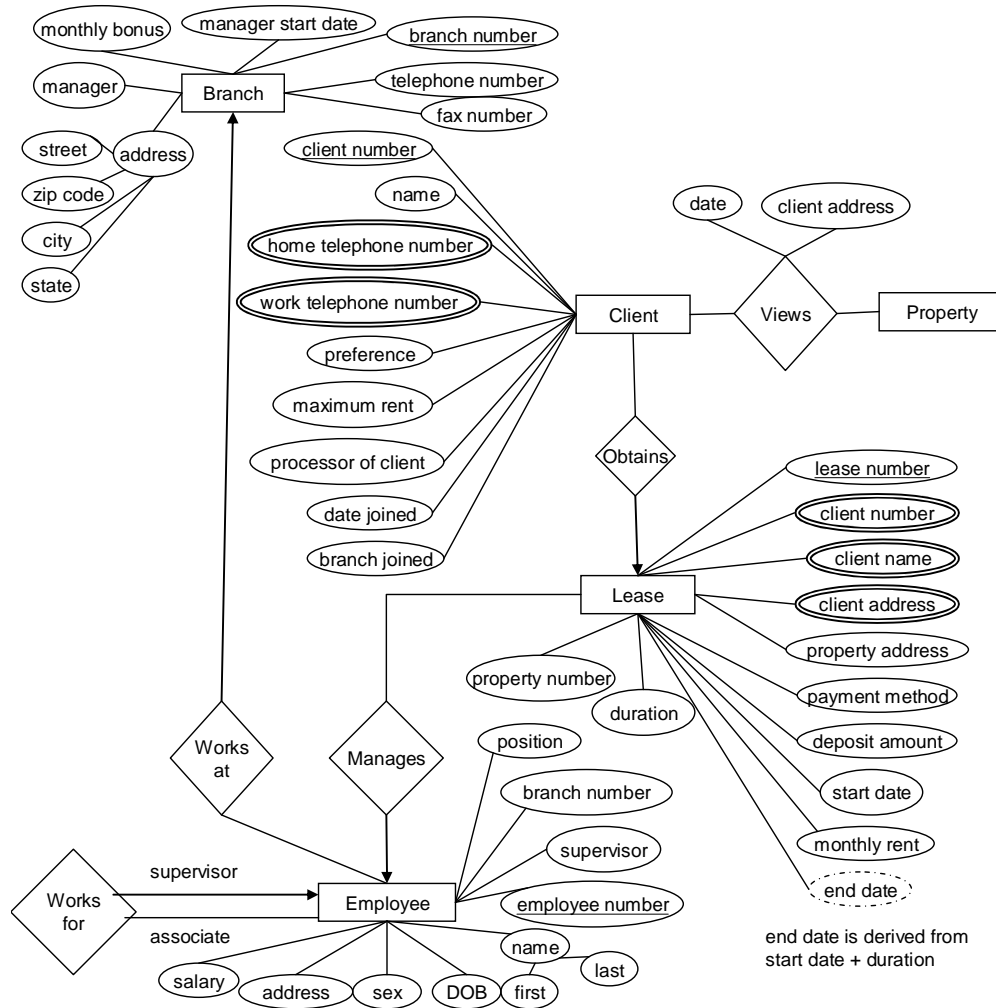
A menu will come up offering the following reports:

- 1 - List the details of branches in a given city
- 2 - Find the total number of branches in a given state
- 3 - List name, position, and salary of all employees at a given branch, ordered by name
- 4 - Find the total number of employees and the sum of their salaries
- 5 - List the number of employees in each position at branches in Baltimore, MD
- 6 - List the name of branch managers, ordered by branch address
- 7 - List the names of associates supervised by a given supervisor
- 8 - List the details of properties in a given city, along with their owner's details, ordered by rent
- 9 - List the details of properties for rent assigned to each associate at a given branch
- 10 - List the details of properties provided by business owners at a given branch
- 11 - Find the total number of properties of each type at all branches
- 12 - List the details of private property owners that provide more than one property for rent
- 13 - List the details of apartments with at least two bedrooms in Baltimore, MD with a monthly rent of at most \$1200
- 14 - List the details of clients registered at a branch, together with their preferences, which have not signed a lease yet
- 15 - List the details of the owner of a given property
- 16 - List the comments made by clients that viewed a given property
- 17 - Find those properties that have been advertised more than the average number of times
- 18 - List the details of leases due to expire next month at a given branch
- 19 - For each state, list the total number of leases with rental duration less than 12 months
- 20 - Find the total current monthly rental income, total monthly management fees, total salaries, as well as the maximum possible monthly rental income and management fee
- 21 - List the details of properties that have not been rented out for the last three months
- 22 - List the details of clients whose preferences match a given property

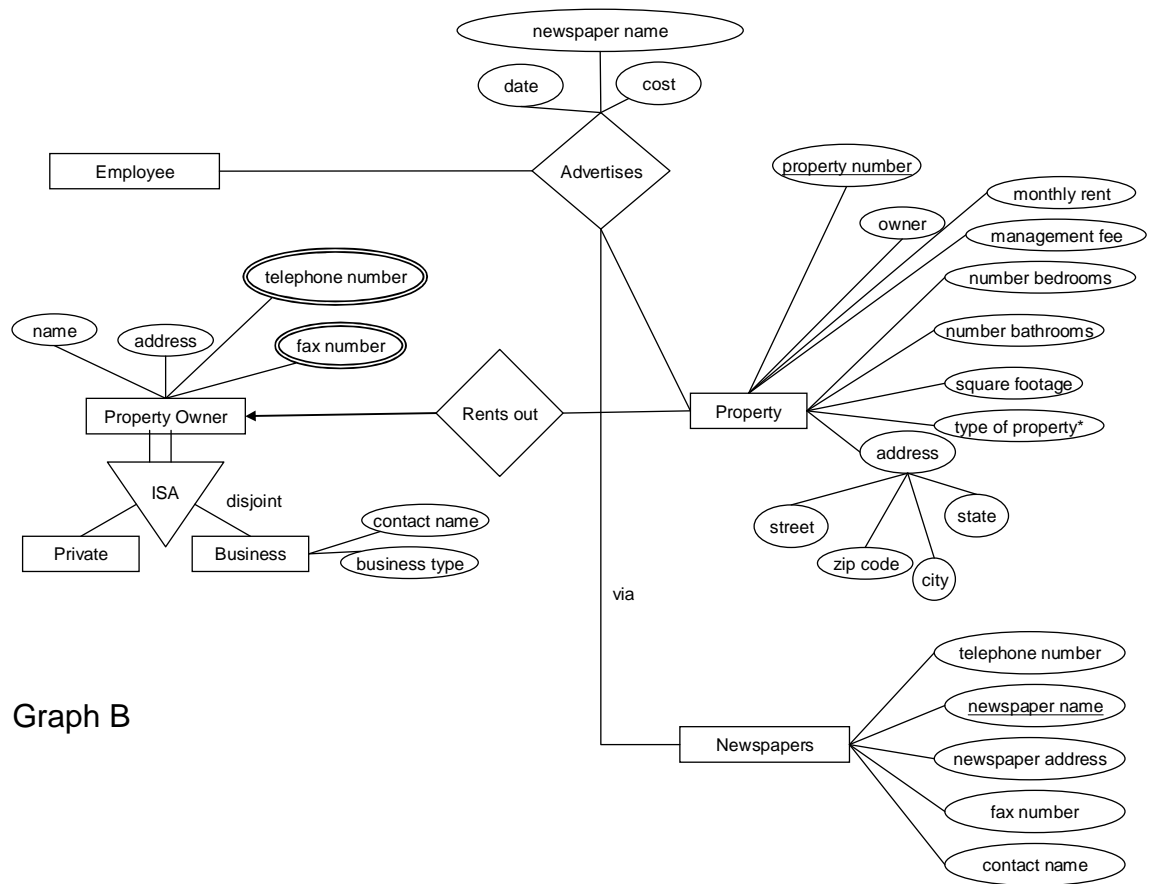
Some of these reports require user input and the user will be prompted for this input.

NOTE: Whenever it asks for a number value, such as salary, it must be an integer in this implementation. Also whenever the prompt specifies input in parentheses, one of the listed options must be used verbatim.

Appendix A



Graph A



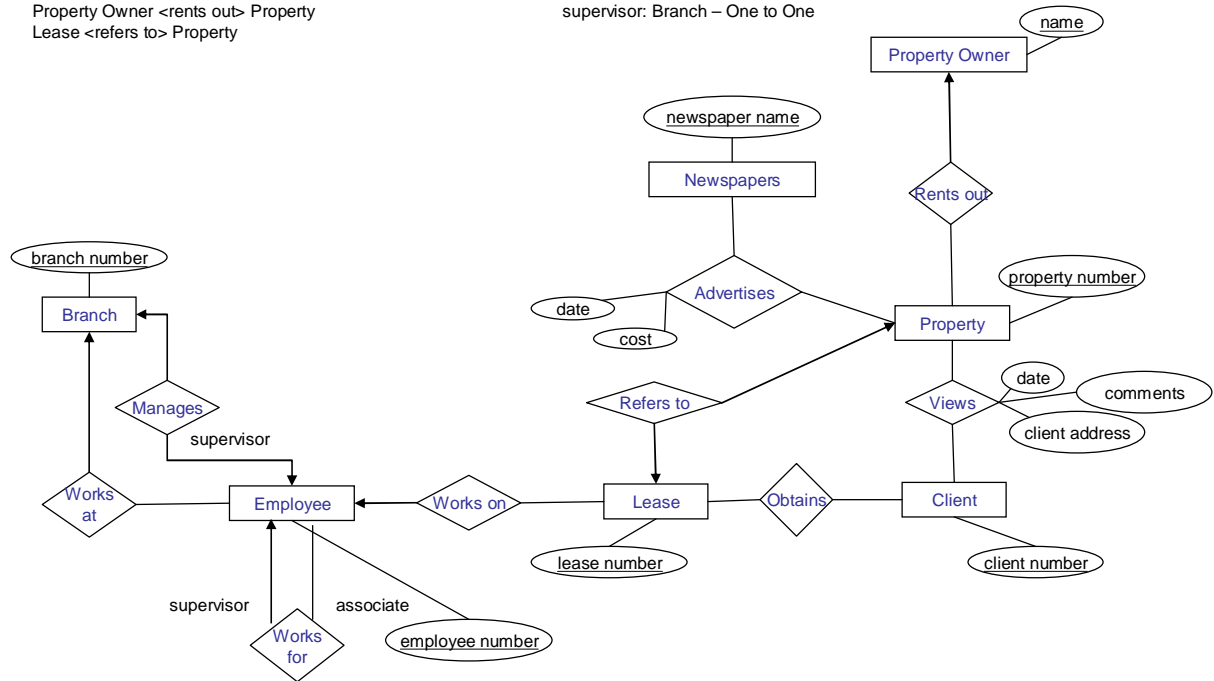
Graph B

*valid: apartment, townhouse, single family house

Appendix B

Employee <works for> Employee: associate works for supervisor
 Employee <works at> Branch
 Employee <manages> Branch: supervisor manages Branch
 Property <advertises> Newspapers
 Employee <works on> Lease
 Client <obtains> Lease
 Client <views> Property
 Property Owner <rents out> Property
 Lease <refers to> Property

Property Owner: Property – One to Many
 Property : Client – Many to Many
 Lease : Property – One to One
 Employee: Lease – One to Many
 Newspapers: Property – Many to Many
 supervisor: associate – One to Many
 Employee: Branch – Many to One
 supervisor: Branch – One to One



Phase B

Appendix C

```
create table Newspaper
(newspaper_name      varchar(32)          not null unique,
tele                char(10)   not null,
addr                varchar(100)         not null,
fax                 char(10)   not null,
contact             char(32)   not null,
primary key (newspaper_name));
```

```
create table Branch
(branch_num          int                not null unique,
manager             varchar(32)         not null unique,
addr_str            varchar(32)         not null,
addr_city           varchar(32)         not null,
addr_st             varchar(16)         not null,
addr_zip            char(5)             not null,
fax                 char(10)   not null,
tele                char(10)   not null,
monthly_bonus       int                 not null,
manager_start       date                not null,
primary key (branch_num),
check (monthly_bonus >= 0));
```

```
create table Client
(client_num          int                not null unique,
name                varchar(32)         not null,
preference          varchar(32)         not null,
max_rent            int                 not null,
employee_name       varchar(32)         not null,
date_joined         date                not null,
branch_joined       int,
primary key (client_num),
foreign key (branch_joined) references Branch(branch_num)
on delete set null);
```

```
create table ClientHomeTele
(client_num          int                not null,
home_tele           char(10)   not null,
primary key (client_num, home_tele),
foreign key (client_num) references Client);
```

```
create table ClientWorkTele
(client_num          int                not null,
work_tele           char(10)   not null,
primary key (client_num, work_tele),
foreign key (client_num) references Client);
```

```
create table Employee
(emp_num            int                not null unique,
position           char(15)   not null,
branch_num         int                not null,
supervisor         varchar(32),
name               varchar(64)         not null,
salary            float(2)   not null,
addr              varchar(100)         not null,
sex               char(1)   not null,
dob              date                not null,
primary key (emp_num),
check (position in ('Associate','Supervisor')),
check (salary >= 0),
foreign key (branch_num) references Branch);
```

```
create table PropOwner
(name              varchar(64)         not null unique,
private           char(3)             not null,
addr              varchar(100)         not null,
contact           varchar(32),
type              varchar(64),
```

```

primary key (name));

create table PropOwnerTele
(name          varchar(64)          not null,
tele          char(10) not null,
primary key (name),
foreign key (name) references PropOwner);

create table PropOwnerFax
(name          varchar(64)          not null,
fax           char(10) not null,
primary key (name),
foreign key (name) references PropOwner);

create table Property
(property_num  int                  not null unique,
owner         varchar(32)          not null,
rent          float(2) not null,
fee           float(2) not null,
bedrooms      smallint not null,
bathrooms     smallint not null,
sq_footage    int                  not null,
type          varchar(32)          not null,
addr_str      varchar(32)          not null,
addr_city     varchar(32)          not null,
addr_st       varchar(16)          not null,
addr_zip      char(5) not null,
primary key (property_num),
check (rent >= 0),
check (fee >= 0),
check (bedrooms > 0),
check (bathrooms > 0),
check (sq_footage > 0),
check (type in ('apartment','townhouse','single family house')),
foreign key (owner) references PropOwner(name));

create table Lease
(lease_num    int                  not null unique,
property_num  int                  not null unique,
addr         varchar(100)          not null,
pay_method    char(10) not null,
deposit       float(2) not null,
startdate     date                  not null,
enddate       date                  not null,
rent          float(2) not null,
primary key (lease_num),
check (deposit >= 0),
check (rent >= 0),
foreign key (property_num) references Property);

create table LeaseClientInfo
(lease_num    int                  not null,
client_name   varchar(32)          not null,
client_addr   varchar(100)          not null,
primary key (client_name, client_addr),
foreign key (lease_num) references Lease);

create table advertises
(property_num  int                  not null,
newspaper_name varchar(32),
the_date      date                  not null,
cost          float(2) not null,
primary key (property_num, newspaper_name),
check (cost >= 0),
foreign key (property_num) references Property,
foreign key (newspaper_name) references Newspaper
on delete set null);

create table views
(client_num    int                  not null unique,

```

```
property_num    int                not null unique,  
the_date        date               not null,  
comments        varchar(200),  
client_addr     varchar(100)       not null,  
primary key (client_num, property_num),  
foreign key (client_num) references Client,  
foreign key (property_num) references Property);
```

create table obtains

```
(lease_num      int                not null,  
client_num      int                not null unique,  
primary key (client_num),  
foreign key (lease_num) references Lease,  
foreign key (client_num) references Client);
```

/* An employee can work on up to 30 Leases */

create table works_on

```
(lease_num      int                not null unique,  
emp_num         int                not null,  
primary key (lease_num),  
foreign key (lease_num) references Lease,  
foreign key (emp_num) references Employee);
```

create table rents

```
(name           varchar(64)        not null,  
property_num    int                not null unique,  
primary key (property_num),  
foreign key (name) references PropOwner,  
foreign key (property_num) references Property);
```

Appendix D

/* Build Entity Tables */

/* Populate Branch table */

```
insert into Branch values ('123','John Daniels','Main St.','Pheonix','Arizona','85749','5204395817','5207893212','1200',
to_date('2005/12/20','YYYY/MM/DD'));
insert into Branch values ('124','Jim Bean','Oak St','Tulsa','Ohio','12345','4109874512','7896540123','1000',
to_date('2004/12/12','YYYY/MM/DD'));
insert into Branch values ('125','Chivas Regal','Palm Ave','Houston','Texas','85746','6104561232','6104561233','5000',
to_date('2003/12/10','YYYY/MM/DD'));
insert into Branch values ('126','Captain Morgan','Peach St','Key
West','Florida','12341','7495236545','7495236544','200', to_date('2006/1/1','YYYY/MM/DD'));
insert into Branch values ('127','Jimmy Dean','Porker Rd','New York','New
York','12346','2124569889','2124569899','550', to_date('2006/2/2','YYYY/MM/DD'));
```

/* Populate Employee table */

```
insert into Employee values ('1231','Supervisor','123','','John Daniels','120000.28','128 Fair St, Pheonix, AZ,
85749','M', to_date('1746/12/8','YYYY/MM/DD'));
insert into Employee values ('1232','Associate','123','John Daniels','John Smith','65000','18 Roanoke, Pheonix,
AZ, 85749','M', to_date('1956/12/8','YYYY/MM/DD'));
insert into Employee values ('1233','Associate','123','John Daniels','Thomas Smith','35000','17 Ro, Pheonix, AZ,
85749','M', to_date('1976/12/8','YYYY/MM/DD'));
insert into Employee values ('1241','Supervisor','124','','Jim Bean','125000','Elm St, Tulsa, Ohio, 12345','M',
to_date('1800/12/9','YYYY/MM/DD'));
insert into Employee values ('1242','Associate','124','Jim Bean','Joe Bob','45500','8th, Tulsa, Ohio, 12345','M',
to_date('1966/4/5','YYYY/MM/DD'));
insert into Employee values ('1243','Associate','124','Jim Bean','Jim Bob','55000','9th, Tulsa, Ohio, 12345','M',
to_date('1965/3/4','YYYY/MM/DD'));
insert into Employee values ('1251','Supervisor','125','','Chivas Regal','150000','45 Cool Std, Houston, TX,
85746','M', to_date('1776/7/8','YYYY/MM/DD'));
insert into Employee values ('1252','Associate','125','Chivas Regal','Jason Marky','45000','5 Cool Std, Houston,
TX, 85746','M', to_date('1776/7/9','YYYY/MM/DD'));
insert into Employee values ('1253','Associate','125','Chivas Regal','Mark Megan','45500','6 Cool Std, Houston,
TX, 85746','M', to_date('1776/7/10','YYYY/MM/DD'));
insert into Employee values ('1261','Supervisor','126','','Captain Morgan','145000','Margaritaville, Key West, FL,
12341','M', to_date('1975/8/8','YYYY/MM/DD'));
insert into Employee values ('1262','Associate','126','Captain Morgan','Josh Dun','45000','46 The St, Key West,
FL, 12341','M', to_date('1981/12/25','YYYY/MM/DD'));
insert into Employee values ('1263','Associate','126','Captain Morgan','Jon McNich','45000','45 The St, Key West,
FL, 12341','M', to_date('1980/12/7','YYYY/MM/DD'));
insert into Employee values ('1271','Supervisor','127','','Jimmy Dean','130000','45 Drunk Blvd, New York, NY,
12346','M', to_date('1965/12/22','YYYY/MM/DD'));
insert into Employee values ('1272','Associate','127','Jimmy Dean','Kevin Mary','68000.25','15th St, New York,
NY, 12346','M', to_date('1966/12/10','YYYY/MM/DD'));
insert into Employee values ('1273','Associate','127','Jimmy Dean','Paul Trip','5.25','12 Loser St, New York, NY,
12346','M', to_date('1976/11/7','YYYY/MM/DD'));
```

/* Populate Newspaper table */

```
insert into Newspaper values ('Baltimore Sun','4104395555','Pratt St, Baltimore, MD,
21250','4104396666','Charles Frasier');
insert into Newspaper values ('Baltimore Gazette','4104375555','Fayette St, Baltimore, MD,
21250','4104376666','James Smith');
insert into Newspaper values ('New York Times','2128884455','1st St, New York, NY, 12346','2128886655','The
Man');
```

/* Populate PropOwner, PropOwnerTele, PropOwnerFax table */

```
insert into PropOwner values ('Owner 1','Yes','North St, Ireland','');
insert into PropOwner values ('Owner 2','No','South St, Poland','George Smith','Rentals');
insert into PropOwnerTele values ('Owner 1','4104569878');
insert into PropOwnerTele values ('Owner 2','4109891212');
insert into PropOwnerFax values ('Owner 1','4104569788');
insert into PropOwnerFax values ('Owner 2','4109892121');
```

/* Populate Proerty table */

```
insert into Property values ('321','Owner 1','450','10','2','5','1100','apartment','Elm St','Pheonix','Arizona','85749');
insert into Property values ('322','Owner 1','450','10','2','5','1100','apartment','North St','Pheonix','Arizona','85749');
insert into Property values ('323','Owner 2','450','10','5','5','3500','townhouse','1 South Blvd','New York','New
York','12346');
```

```

insert into Property values ('324','Owner 2','450','10','6','5','5600','single family house','11 South Blvd','New York','New York','12346');
insert into Property values ('325','Owner 2','450','10','6','5','10000','single family house','12 South Blvd','New York','New York','12346');

```

```

/* Populate Client, ClientHomeTele, ClientWorkTele table */

```

```

insert into Client values ('1','Buckaroo Bonzai','apartment','700','John Smith', to_date('2006/1/1','YYYY/MM/DD'),'123');
insert into Client values ('2','New Jersey','townhouse','600','Joe Bob', to_date('2005/4/5','YYYY/MM/DD'),'124');
insert into Client values ('3','Boyd Sheldon','single family house','1100','Kevin Mary',
to_date('2005/7/1','YYYY/MM/DD'),'127');
insert into Client values ('4','Foster Sheldon','apartment','4500','Jon McNich', to_date('2006/1/4','YYYY/MM/DD'),'126');
insert into ClientHomeTele values ('1','2221234565');
insert into ClientHomeTele values ('2','2223214444');
insert into ClientHomeTele values ('3','4444444444');
insert into ClientHomeTele values ('4','5546546654');
insert into ClientWorkTele values ('1','2221234488');
insert into ClientWorkTele values ('2','222222100');
insert into ClientWorkTele values ('3','5444444444');
insert into ClientWorkTele values ('4','6666666666');

```

```

/* Populate Lease, LeaseClientInfo table */

```

```

insert into Lease values ('789','321','Elm St, Pheonix, Arizona, 85749','Check','5500',
to_date('2006/8/13','YYYY/MM/DD'), to_date('2007/1/1','YYYY/MM/DD'),'450.00');
insert into Lease values ('790','322','North St, Pheonix, Arizona, 85749','Cash','1200',
to_date('2006/8/12','YYYY/MM/DD'), to_date('2007/2/2','YYYY/MM/DD'),'450.00');
insert into Lease values ('791','323','1 South Blvd, New York, New York, 12346','Cash','6500',
to_date('2006/8/11','YYYY/MM/DD'), to_date('2007/3/5','YYYY/MM/DD'),'450.00');
insert into LeaseClientInfo values ('789','Buckaroo Bonzai','Jason Ave, Pheonix, Arizona, 85749');
insert into LeaseClientInfo values ('790','New Jersey','1st St, Tucson, Arizona, 85745');
insert into LeaseClientInfo values ('791','Foster Sheldon','BackWoords, West Virginia');

```

```

/* Build Initial Relation Tables */

```

```

/* Populate advertises table */

```

```

insert into advertises values ('321','Baltimore Sun', to_date('2007/5/6','YYYY/MM/DD'),'12.00');
insert into advertises values ('321','Baltimore Gazette', to_date('2007/5/5','YYYY/MM/DD'),'15.00');
insert into advertises values ('321','New York Times', to_date('2007/5/4','YYYY/MM/DD'),'17.00');
insert into advertises values ('322','Baltimore Gazette', to_date('2007/5/5','YYYY/MM/DD'),'15.00');
insert into advertises values ('322','New York Times', to_date('2007/5/5','YYYY/MM/DD'),'15.00');
insert into advertises values ('323','New York Times', to_date('2007/5/3','YYYY/MM/DD'),'15.00');

```

```

/* Populate views table */

```

```

insert into views values ('3','325', to_date('2006/8/15','YYYY/MM/DD'),'Potential Buy','52nd Blvd, Queens, New York, 12346');

```

```

/* Populate obtains table */

```

```

insert into obtains values ('789','1');
insert into obtains values ('790','2');
insert into obtains values ('791','4');

```

```

/* Populate works_on table */

```

```

insert into works_on values ('789','1232');
insert into works_on values ('790','1242');
insert into works_on values ('791','1262');

```

```

/* Populate rents table */

```

```

/* This table may be redundant */

```

```

insert into rents values ('Owner 1','321');
insert into rents values ('Owner 1','322');
insert into rents values ('Owner 2','323');
insert into rents values ('Owner 2','324');
insert into rents values ('Owner 2','325');

```

Appendix E

delete from advertises;
drop table advertises;

delete from views;
drop table views;

delete from obtains;
drop table obtains;

delete from works_on;
drop table works_on;

delete from rents;
drop table rents;

delete from LeaseClientInfo;
drop table LeaseClientInfo;

delete from Lease;
drop table Lease;

delete from PropOwnerTele;
drop table PropOwnerTele;

delete from PropOwnerFax;
drop table PropOwnerFax;

delete from Property;
drop table Property;

delete from PropOwner;
drop table PropOwner;

delete from Employee;
drop table Employee;

delete from ClientHomeTele;
drop table ClientHomeTele;

delete from ClientWorkTele;
drop table ClientWorkTele;

delete from Client;
drop table Client;

delete from Newspaper;
drop table Newspaper;

delete from Branch;
drop table Branch;

Appendix F

1. select * from Branch where addr_city = 'Tulsa';
2. select count(branch_num) from Branch where addr_st = 'Ohio';
3. select name,position,salary from Employee where branch_num = 123 order by name;
4. select sum(salary), count(emp_num) from Employee;
5. select count(emp_num),position from Employee where Employee.branch_num in
(select branch_num from Branch where Branch.addr_city = 'New York' and Branch.addr_st = 'New
York') group by position;
6. select manager from Branch order by addr_st,addr_city,addr_str;
7. select name from Employee where position = 'Associate' and supervisor = 'John Daniels';
8. select * from Property,PropOwner,PropOwnerTele,PropOwnerFax where
Property.owner = PropOwner.name and PropOwner.name = PropOwnerTele.name and
PropOwner.name = PropOwnerFax.name and Property.addr_city = 'Pheonix' order by Property.rent;
9. select * from Property where property_num in
(select property_num from Employee,Lease,works_on where Employee.emp_num =
works_on.emp_num and works_on.lease_num = Lease.lease_num and Employee.branch_num = 123);
10. select * from Property where property_num in
(select rents.property_num from PropOwner,rents,Lease,works_on,Employee where
PropOwner.name = rents.name and
rents.property_num = Lease.property_num and
Lease.lease_num = works_on.lease_num and
works_on.emp_num = Employee.emp_num and
Employee.branch_num = 126 and
PropOwner.private = 'No');
11. select count(property_num),type from Property group by type;
- 12.
13. select * from Property where type = 'apartment' and bedrooms >= 2 and addr_city = 'Baltimore'
and addr_st = 'MD' and rent <= 1200;
14. select * from Client where client_num not in (select client_num from obtains);
15. select * from PropOwner where name in (select name from rents where property_num = 321);
16. select comments from views where property_num = 325;
- 17.
18. select * from Lease where enddate < add_months(sysdate,1) and lease_num in
(select lease_num from works_on,Employee where works_on.emp_num = Employee.emp_num and
Employee.branch_num = 123);
19. select count(lease_num),addr_st from Lease,Property where Lease.property_num = Property.property_num
and enddate - startdate < 12 * 31 group by Property.addr_st;
20. select sum(rent),sum(fee) from Property where property_num in (select property_num from Lease);
select sum(salary) from (select distinct emp_num,salary from Employee);
select sum(rent),sum(fee) from Property;
21. select * from Lease,Property where enddate <= add_months(sysdate,-3) and Lease.property_num =
Property.property_num
22. select * from Client,ClientHomeTele,ClientWorkTele where Client.client_num = ClientHomeTele.client_num and
Client.client_num = ClientWorkTele.client_num and preference in
(select type from Property where Property.property_num = 321);