

Object-Based Databases

Solutions to Practice Exercises

9.1 For this problem, we use table inheritance. We assume that **MyDate**, **Color** and **DriveTrainType** are pre-defined types.

```
create type Vehicle  
  (vehicle_id integer,  
   license_number char(15),  
   manufacturer char(30),  
   model char(30),  
   purchase_date MyDate,  
   color Color)
```

```
create table vehicle of type Vehicle
```

```
create table truck  
  (cargo_capacity integer)  
under vehicle
```

```
create table sportsCar  
  (horsepower integer  
   renter_age_requirement integer)  
under vehicle
```

```
create table van  
  (num_passengers integer)  
under vehicle
```

```

create table offRoadVehicle
  (ground_clearance real
   driveTrain DriveTrainType)
under vehicle

```

- 9.2 a. No Answer.
b. Queries in SQL:1999.

i. Program:

```

select ename
from emp as e, e.ChildrenSet as c
where 'March' in
  (select birthday.month
   from c
  )

```

ii. Program:

```

select e.ename
from emp as e, e.SkillSet as s, s.ExamSet as x
where s.type = 'typing' and x.city = 'Dayton'

```

iii. Program:

```

select distinct s.type
from emp as e, e.SkillSet as s

```

- 9.3 a. The corresponding SQL:1999 schema definition is given below. Note that the derived attribute *age* has been translated into a method.

```

create type Name
  (first_name varchar(15),
   middle_initial char,
   last_name varchar(15))
create type Street
  (street_name varchar(15),
   street_number varchar(4),
   apartment_number varchar(7))
create type Address
  (street Street,
   city varchar(15),
   state varchar(15),
   zip_code char(6))
create table customer
  (name Name,
   customer_id varchar(10),
   address Address,
   phones char(7) array[10],
   dob date)

```

```
method integer age()
```

```
b. create function Name (f varchar(15), m char, l varchar(15))
returns Name
begin
  set first_name = f;
  set middle_initial = m;
  set last_name = l;
end
create function Street (sname varchar(15), sno varchar(4), ano varchar(7))
returns Street
begin
  set street_name = sname;
  set street_number = sno;
  set apartment_number = ano;
end
create function Address (s Street, c varchar(15), sta varchar(15), zip varchar(6))
returns Address
begin
  set street = s;
  set city = c;
  set state = sta;
  set zip_code = zip;
end
```

- 9.4 a. The schema definition is given below. Note that backward references can be added but they are not so important as in OODBS because queries can be written in SQL and joins can take care of integrity constraints.

```
create type Employee
(person_name varchar(30),
street varchar(15),
city varchar(15))
create type Company
(company_name varchar(15),
city varchar(15))
create table employee of Employee
create table company of Company
create type Works
(person ref(Employee) scope employee,
comp ref(Company) scope company,
salary int)
create table works of Works
create type Manages
(person ref(Employee) scope employee,
(manager ref(Employee) scope employee))
```

create table *manages* of *Manages*

- b. i. **select** *comp*– *>name*
from *works*
group by *comp*
having count(*person*) \geq **all**(**select count**(*person*)
from *works*
group by *comp*)
- ii. **select** *comp*– *>name*
from *works*
group by *comp*
having sum(*salary*) \leq **all**(**select sum**(*salary*)
from *works*
group by *comp*)
- iii. **select** *comp*– *>name*
from *works*
group by *comp*
having avg(*salary*) $>$ (**select avg**(*salary*)
from *works*
where *comp*– *>company_name*="First Bank Corporation")
- 9.5 a. A computer-aided design system for a manufacturer of airplanes:
 An OODB system would be suitable for this. That is because CAD requires complex data types, and being computation oriented, CAD tools are typically used in a programming language environment needing to access the database.
- b. A system to track contributions made to candidates for public office:
 A relational system would be apt for this, as data types are expected to be simple, and a powerful querying mechanism is essential.
- c. An information system to support the making of movies:
 Here there will be extensive use of multimedia and other complex data types. But queries are probably simple, and thus an object relational system is suitable.
- 9.6 An entity is simply a collection of variables or data items. An object is an encapsulation of data as well as the methods (code) to operate on the data. The data members of an object are directly visible only to its methods. The outside world can gain access to the object's data only by passing pre-defined messages to it, and these messages are implemented by the methods.