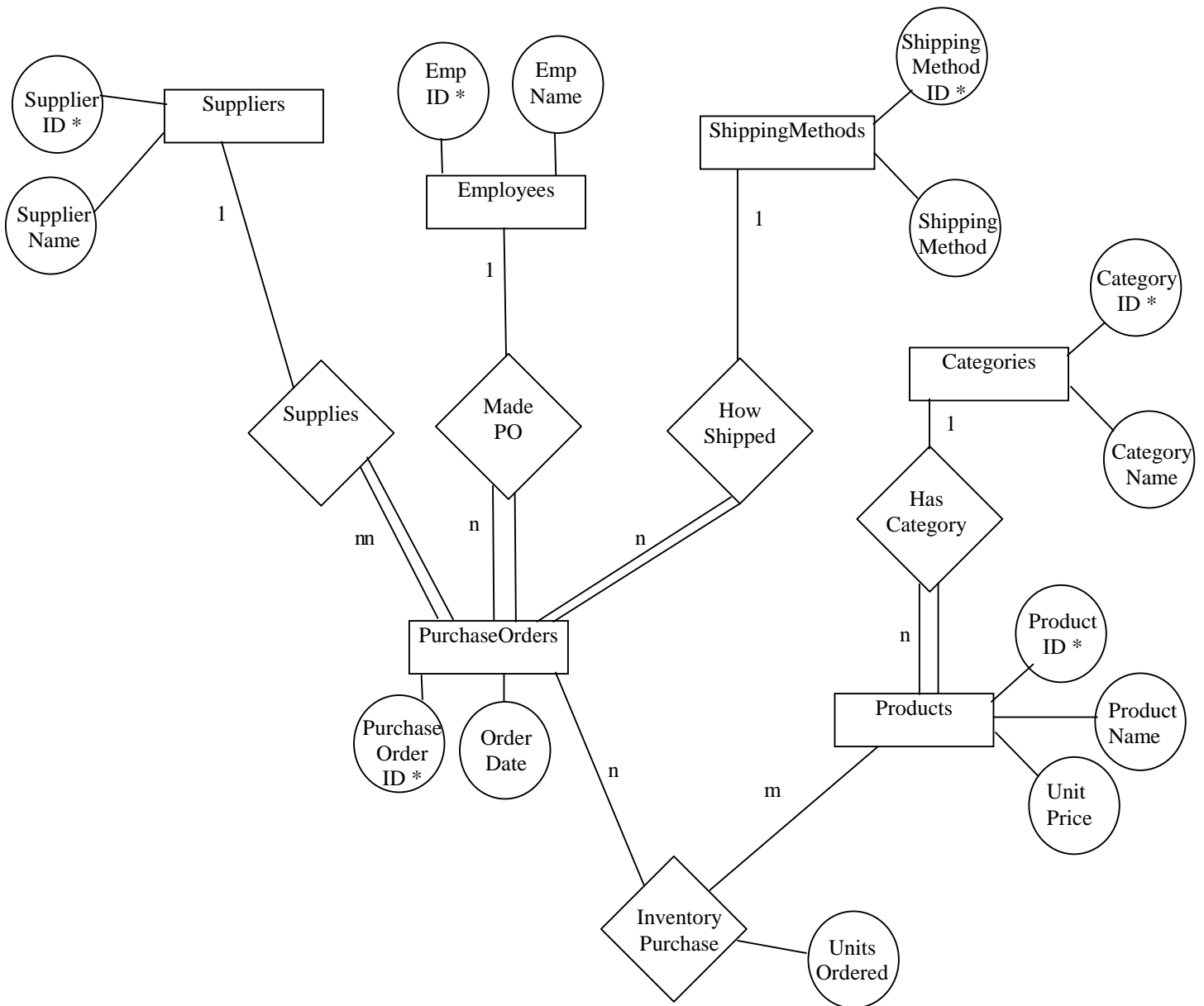


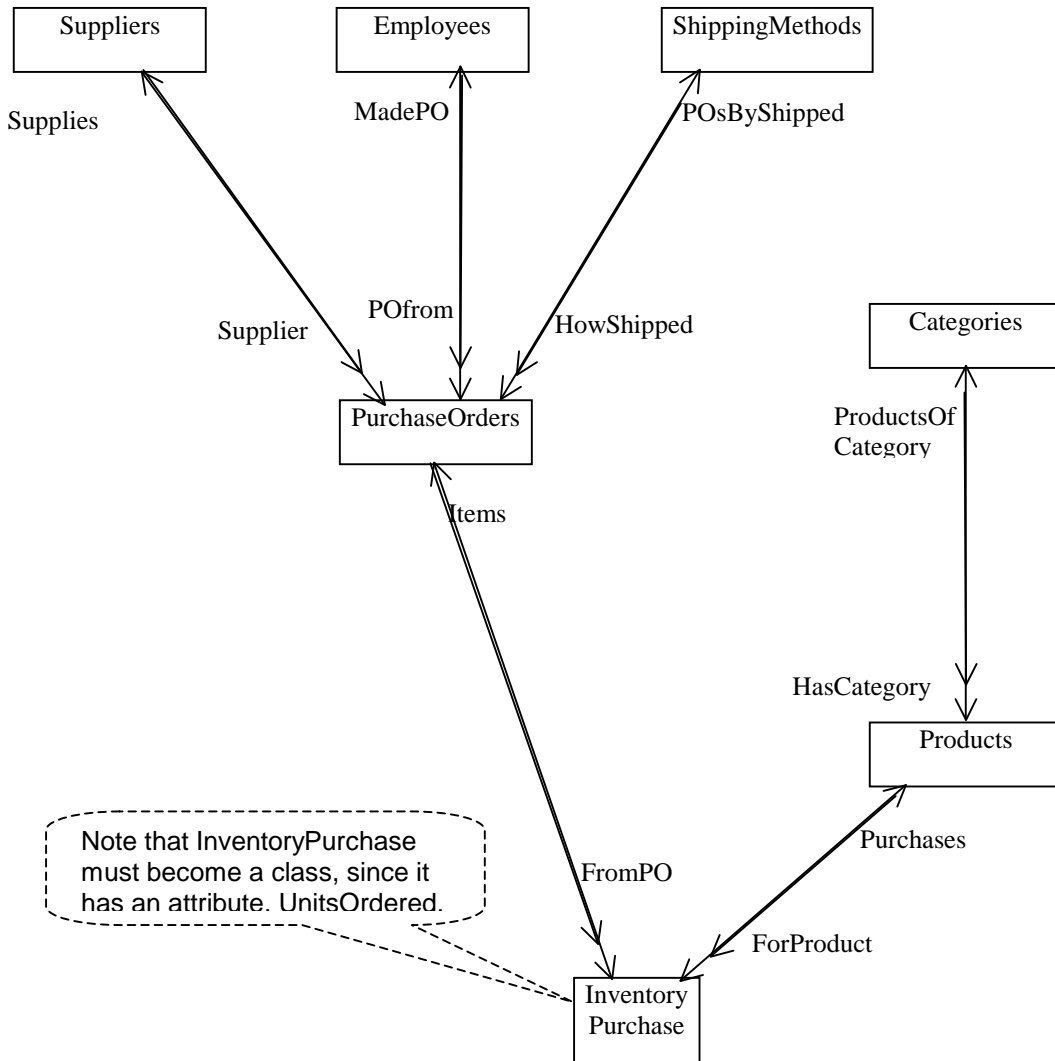
Consider the following scenario:

The XYZ Department Store has had an expensive database design consultant design an entity-relationship database to keep track of its inventory. The database includes products and their product category. Purchases from suppliers are specified through purchase orders. A purchase order specifies one or more products to be purchased. It also records the employee making the order as well as the shipping method to be used to ship the purchase.

An extended entity-relationship diagram for this database is shown below:



- (20 points) Translate the EER diagram shown on Page 1 to the graphical notation used in Elmasri & Navathe, Figure 12.5, p. 400.



(20points) Translate the EER diagram shown on Page 1 to ODL.

```
class Supplier
(extent Suppliers
key SupplierID)
{ attribute string SupplierID;
  attribute string SupplierName;
  relationship set <PurchaseOrder> Supplies
    inverse PurchaseOrder::Supplier;
};

class Employee
(extent Employees
key EmpID)
{ attribute integer EmpID;
  attribute string EmpName;
  relationship set <PurchaseOrder> MadePO
    inverse PurchaseOrder::POfrom;
};

class ShippingMethod
(extent ShippingMethods
key ShippingMethodID)
{ attribute string ShippingMethodID;
  attribute string ShippingMethodName;
  relationship set <PurchaseOrder> POsShipped
    inverse PurchaseOrder::HowShipped;
};

class PurchaseOrder
(extent PurchaseOrders
key PurchaseOrderID)
{ attribute integer PurchaseOrderID;
  attribute date OrderDate;
  relationship Supplier Supplier
    inverse Supplier::Supplies;
  relationship Employee POfrom
    inverse Employee::MadePO;
  relationship ShippingMethod HowShipped
    inverse ShippingMethod::POsByShipped;
  relationship set <InventoryPurchase> Items
    inverse InventoryPurchase::FromPO;
};

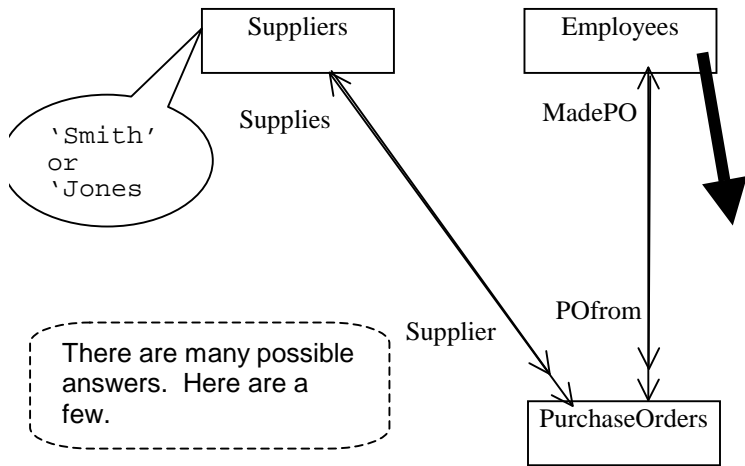
class Category
(extent Categories
key CategoryID)
{ attribute integer CategoryID;
  attribute string CategoryName;
  relationship set <Products> ProductsOfCategory
    inverse Products::HasCategory;
};

class Product
(extent Products
key ProductID)
{ attribute integer ProductID;
  attribute string ProductName;
  attribute integer UnitPrice;
  relationship Category HasCategory
    inverse Category::ProductsOfCategory;
  relationship set <InventoryPurchase> Purchases
    inverse InventoryPurchase::ForProduct;
};

class InventoryPurchase
(extent InventoryPurchases)
{ attribute integer UnitsOrdered;
  relationship Product ForProduct
    inverse Product::Purchases;
  relationship PurchaseOrder FromPO
    inverse PurchaseOrder::ForProduct;
};
```

Translate the following queries into *OQL*.

2. (15 points) List the names of all employees who have placed an order with **either** supplier Smith **or** with supplier Jones.



```
select distinct e.EmpName
from s in Suppliers,
e in s. Supplies.Pofrom
where s.SupplierName = 'Smith'
or s.SupplierName = 'Jones'
```

```
select distinct e.EmpName
from e in Employee
where exists
    po in e.MadePO,
    s in po.Supplier
: s.SupplierName = 'Smith'
or s.SupplierName = 'Jones'
```

```
select distinct e.EmpName
from e in Employee
where exists
    (po in e.MadePO,
    : po.Supplier.SupplierName = 'Smith'
or po.Supplier.SupplierName = 'Jones')
```

```
select distinct e.EmpName
from s in Suppliers,
po in s.Supplies,
e in po.Pofrom
where s.SupplierName = 'Smith'
or s.SupplierName = 'Jones'
```

```
select e.EmpName
from e in Employees
where 'Smith' in e.MadePO.Supplier.SupplierName
or 'Jones' in e.MadePO.Supplier.SupplierName
```

Note that this is a set.

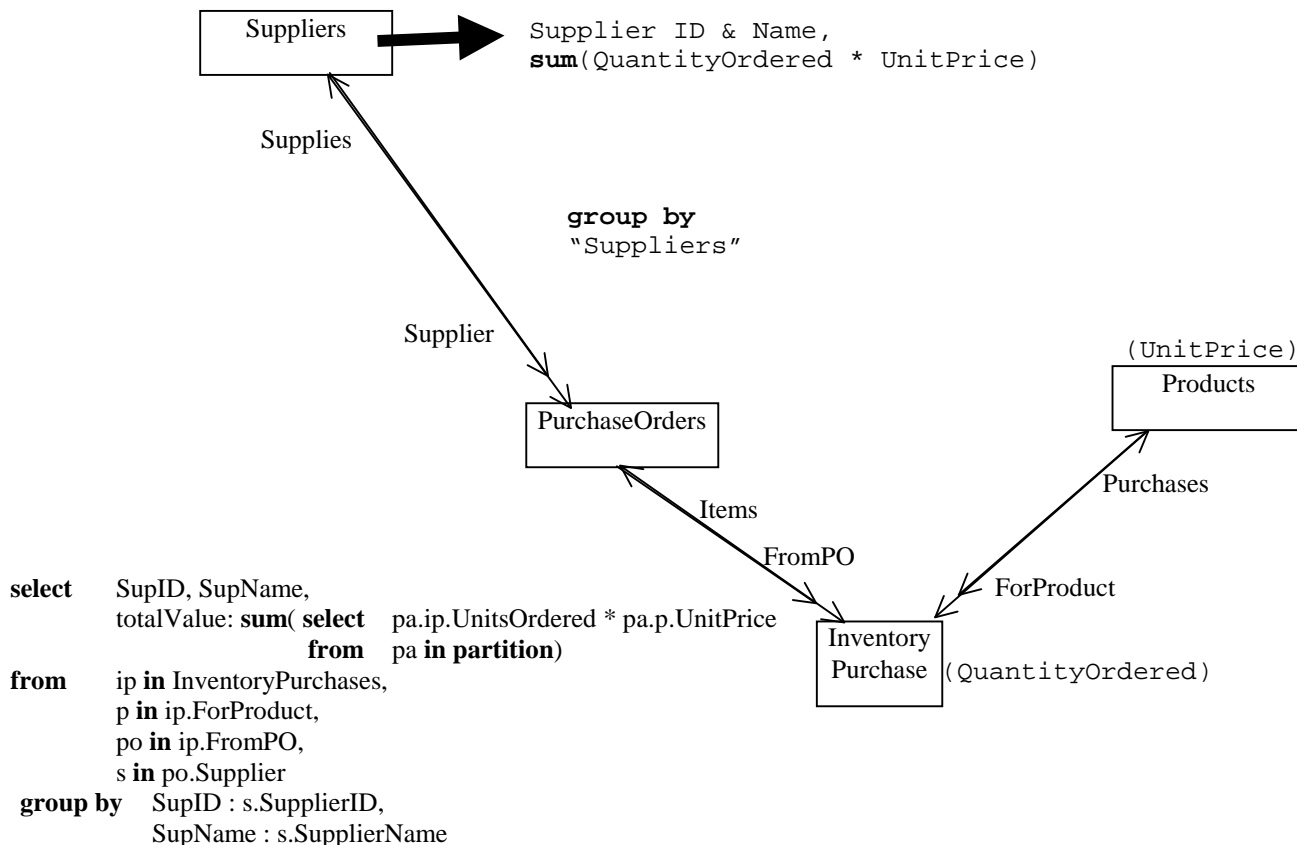
3. (15 points) List the names of all employees who have placed an order with **both** supplier Smith **and** with supplier Jones.

```
select distinct e.EmpName
from e in Employees,
po1 in e.MadePO, po2 in e.MadePO,
s1 in po1.Supplier, s2 in po2. Supplier
where s1.SupplierName = 'Smith'
and s2.SupplierName = 'Jones'
```

```
select distinct e.EmpName
from e in Employee
where exists
    (po in e.MadePO,
    s in po.Supplier
: s.SupplierName = 'Smith')
and exists
    (po in e.MadePO,
    s in po.Supplier
: s.SupplierName = 'Jones')
```

There must be two **different** instances of PurchaseOrders and Suppliers, one for 'Smith', the other for 'Jones'.

4. (15 points) For each supplier, list the supplier's ID, name, and total value of products purchased from that supplier. Note that the value of a product in a particular order is the so-called Extended Price, i.e. UnitsOrdered \* UnitPrice.



5. (15 points) List the names of all employees who placed at least one order with every supplier.

