```
--1
CREATE TABLE MyDept (
DeptID NUMBER(5) NOT NULL,
Dname VARCHAR2(20) NOT NULL,
PRIMARY KEY(DeptID)
);

--2
INSERT INTO MYDEPT
VALUES(1, 'Sales');

INSERT INTO MYDEPT
VALUES(2, 'Accounting');

INSERT INTO MYDEPT
VALUES(3, 'Marketing');

INSERT INTO MYDEPT
VALUES(4, 'Human Resources');

--3
CREATE TABLE MYEMP
(
    EMPID NUMBER(5) NOT NULL,
    FNAME VARCHAR(30) NOT NULL,
    LNAME VARCHAR(30) NOT NULL,
    DEPTID NUMBER(5),
    SAL NUMBER(8, 2) NOT NULL,
    HIREDATE DATE NOT NULL,
    PRIMARY KEY(EMPID),
    FOREIGN KEY(DEPTID) REFERENCES MYDEPT(DEPTID)
);

--4
INSERT INTO MYEMP
VALUES(21, 'Jane', 'Smith', 1, 12000, TO_DATE('03-10-2016', 'dd-mm-
YYYY'));

--5
-- ..@/populateExamTables.sql

--6
SELECT FNAME || ' ' || LNAME AS "Full Name",
       SYSDATE - HIREDATE AS "Time with company"
FROM MYEMP;

--7

SELECT FNAME || ' ' || LNAME AS "Full Name"
FROM MYEMP
WHERE DEPTID IS NULL;
```

```
-- 8

SELECT FNAME
FROM MYEMP E, MYDEPT D
WHERE E.DEPTID = D.DEPTID AND D.DNAME = 'Accounting';

-- 9
SELECT EMPID, Dname
FROM MyEmp E, MYDEPT D
WHERE (SYSDATE - HIREDATE) / 365 >= 1 and E.DEPTID = D.DEPTID;
;

-- 10
SELECT FNAME || ' ' || LNAME AS "Full Name",
       12 * SAL AS "Yearly Salary",
       D.Dname AS "Department"
FROM MYEMP E1, MyDept D
WHERE SAL >
    (SELECT AVG(SAL) FROM MYEMP E2 WHERE E2.DEPTID = E1.DEPTID) AND
    E1.DeptID = D.DeptID;

-- 11

SELECT P.PurchaseID, SUM(PD.QuantityPurchased * F.UnitPrice)
FROM MyPurchase P, PurchaseDetails PD, MyFurniture F
WHERE P.PurchaseID = PD.PurchaseID and PD.FurnitureID = F.FurnitureID
GROUP BY P.PurchaseID
ORDER BY 2;

-- 12
SELECT C.CustID, COUNT(P.PurchaseID)
FROM MyCustomer C, MyPurchase P
WHERE C.CustID = P.CustID
GROUP BY C.CustID
HAVING COUNT(P.PurchaseID) >= 3;

-- 13
CREATE OR REPLACE VIEW GOODSALESPEOPLE AS
SELECT FNAME || ' ' || LNAME AS "Full name"
FROM MYEMP E, MYPURCHASE P
WHERE E.EMPID = P.EMPID
GROUP BY FNAME || ' ' || LNAME
HAVING COUNT(P.PurchaseID) > 2;

-- 14
CREATE OR REPLACE PROCEDURE FurnitureSale (productCode in NUMBER,
qtySold in NUMBER)
AS
 BEGIN
  UPDATE MyFurniture
  SET QuantityInStock = QuantityInStock - qtySold
  WHERE FurnitureID = productCode;
 END;

BEGIN
 FurnitureSale(1,2);
END;
/

-- 15
CREATE TABLE MusicCatalog (
ArtistName VARCHAR2(25),
CD SYS.XMLTYPE,
CreationDate Date,
PRIMARY KEY(ArtistName)
);

INSERT INTO MusicCatalog values
('Eros Ramazzotti', SYS.XMLTYPE.createXML(
'<ENTRY>
  <TITLE>Eros</TITLE>
  <COUNTRY>EU</COUNTRY>
  <COMPANY>BMG</COMPANY>
  <PRICE>9.90</PRICE>
  <YEAR>1997</YEAR>
</ENTRY>'), SYSDATE);

INSERT INTO MusicCatalog values
('Joel Hoel', SYS.XMLTYPE.createXML(
'<ENTRY>
  <TITLE>Soulsville</TITLE>
  <COUNTRY>Norway</COUNTRY>
  <COMPANY>WEA</COMPANY>
  <PRICE>7.90</PRICE>
  <YEAR>1996</YEAR>
</ENTRY>'), SYSDATE);

SELECT m.CD.getClobVal()
FROM MusicCatalog m
WHERE ArtistName = 'Joel Hoel';

SELECT m.ArtistName, m.CD.extract('/ENTRY/TITLE).getStringVal()
FROM MusicCatalog m;

SELECT m.ArtistName
FROM MusicCatalog m
WHERE m.CD.existsNode('/ENTRY[COUNTRY = "Norway"]') = 1;

SELECT m.ArtistName,
m.CD.extract('/ENTRY/COMPANY/text()').getStringVal()
FROM MusicCatalog m
WHERE m.CD.existsNode('/ENTRY[PRICE = "7.90"]') = 1;
  /*students can assume price is stored as a string*/
```