CREATE TABLE vendor
  ( vendorid     CHAR(2)  NOT NULL,
    vendorname  VARCHAR(25)   NOT NULL,
    PRIMARY KEY (vendorid) );

CREATE TABLE category
  ( categoryid     CHAR(2)   NOT NULL,
    categoryname   VARCHAR(25)   NOT NULL,
    PRIMARY KEY (categoryid) );

CREATE TABLE product
  ( productid    CHAR(3)   NOT NULL,
    productname  VARCHAR(25)   NOT NULL,
    productprice  NUMERIC (7,2)  NOT NULL,
    vendorid     CHAR(2)   NOT NULL,
    categoryid    CHAR(2)   NOT NULL,
    PRIMARY KEY (productid),
    FOREIGN KEY (vendorid) REFERENCES vendor(vendorid),
    FOREIGN KEY (categoryid) REFERENCES category(categoryid) );

CREATE TABLE region
  ( regionid     CHAR   NOT NULL,
    regionname  VARCHAR(25)   NOT NULL,
    PRIMARY KEY (regionid) );

CREATE TABLE store
  ( storeid     VARCHAR(3)   NOT NULL,
    storezip  CHAR(5)   NOT NULL,
    regionid     CHAR   NOT NULL,
    PRIMARY KEY (storeid),
    FOREIGN KEY (regionid) REFERENCES region(regionid) );

CREATE TABLE customer
  ( customerid   CHAR(7)   NOT NULL,
    customername   VARCHAR(15)   NOT NULL,
    customerzip   CHAR(5)   NOT NULL,
    PRIMARY KEY (customerid) );

CREATE TABLE salestransaction
  ( tid      VARCHAR(8)   NOT NULL,
    customerid  CHAR(7)   NOT NULL,
    storeid  VARCHAR(3)   NOT NULL,
    tdate   DATE   NOT NULL,
    PRIMARY KEY (tid),
    FOREIGN KEY (customerid) REFERENCES customer(customerid),
    FOREIGN KEY (storeid) REFERENCES store(storeid));

CREATE TABLE soldvia
  ( productid    CHAR(3)   NOT NULL,
    tid      VARCHAR(8)   NOT NULL,
    noofitems  INT   NOT NULL,
    PRIMARY KEY (productid, tid),
    FOREIGN KEY (productid) REFERENCES product(productid),
    FOREIGN KEY (tid) REFERENCES salestransaction(tid) );
DROP TABLE sequence;

ZAGI database – VALID:

INSERT INTO vendor VALUES ('PG','Pacifica Gear');
INSERT INTO vendor VALUES ('MK','Mountain King');

INSERT INTO category VALUES ('CP','Camping');
INSERT INTO category VALUES ('FW','Footwear');

INSERT INTO product VALUES ('1X1','Zzz Bag',100,'PG','CP');
INSERT INTO product VALUES ('2X2','Easy Boot',70,'MK','FW');
INSERT INTO product VALUES ('3X3','Cosy Sock',15,'MK','FW');
INSERT INTO product VALUES ('4X4','Dura Boot',90,'PG','FW');
INSERT INTO product VALUES ('5X5','Tiny Tent',150,'MK','CP');
INSERT INTO product VALUES ('6X6','Biggy Tent',250,'MK','CP');

INSERT INTO region VALUES ('C','Chicagoland');
INSERT INTO region VALUES ('T','Tristate');

INSERT INTO store VALUES ('S1','60600','C');
INSERT INTO store VALUES ('S2','60605','C');
INSERT INTO store VALUES ('S3','35400','T');

INSERT INTO customer VALUES ('1-2-333','Tina','60137');
INSERT INTO customer VALUES ('2-3-444','Tony','60611');
INSERT INTO customer VALUES ('3-4-555','Pam ','35401');

INSERT INTO salestransaction VALUES ('T111','1-2-333','S1','01/Jan/2013');
INSERT INTO salestransaction VALUES ('T222','2-3-444','S2','01/Jan/2013');
INSERT INTO salestransaction VALUES ('T333','1-2-333','S3','02/Jan/2013');
INSERT INTO salestransaction VALUES ('T444','3-4-555','S3','02/Jan/2013');
INSERT INTO salestransaction VALUES ('T555','2-3-444','S3','02/Jan/2013');

INSERT INTO soldvia VALUES ('1X1','T111',1);
INSERT INTO soldvia VALUES ('2X2','T222',1);
INSERT INTO soldvia VALUES ('3X3','T333',5);
INSERT INTO soldvia VALUES ('4X4','T444',1);
INSERT INTO soldvia VALUES ('2X2','T444',2);
INSERT INTO soldvia VALUES ('4X4','T555',4);
INSERT INTO soldvia VALUES ('5X5','T555',2);
INSERT INTO soldvia VALUES ('6X6','T555',1);
Query 1:  
```
SELECT  
    productid, productname, productprice, vendorid, categoryid
FROM
    product;
```

Query 1a:  
```
SELECT  
    *
FROM
    product;
```

Query 2:  
```
SELECT  
    productname, productid, vendorid, categoryid, productprice
FROM
    product;
```

Query 3:  
```
SELECT  
    productid, productprice
FROM
    product;
```

Query 3a:  
```
SELECT  
    productid, productprice, productprice * 1.1
FROM
    product;
```

Query 4:  
```
SELECT  
    productid, productname, vendorid, productprice
FROM
    product
WHERE
    productprice > 100;
```

Query 5:  
```
SELECT  
    productid, productname, vendorid, productprice
FROM
    product
WHERE
    productprice <= 110 AND categoryid = 'FW';
```

Query 6:  
```
SELECT  
    vendorid
FROM
    product;
```

Query 7:  
```
SELECT DISTINCT  
    vendorid
FROM
    product;
```

Query 8:  
```
SELECT  
    productid, productname, categoryid, productprice
FROM
    product
WHERE
    categoryid = 'FW'
ORDER BY
    productprice;
```

Query 9:  
```
SELECT  
    productid, productname, categoryid, productprice
FROM
    product
WHERE
    categoryid = 'FW'
ORDER BY
    productprice DESC;
```

Query 10:  
```
SELECT  
    productid, productname, categoryid, productprice
FROM
    product
ORDER BY
    categoryid, productprice;
```

Query 11:  
```
SELECT  
    *
FROM
    product
WHERE
    productname LIKE '%Boot%';
```
Query 12:
```
SELECT AVG(productprice)
FROM product;
```

Query 13:
```
SELECT COUNT(*)
FROM product;
```

Query 14:
```
SELECT COUNT(DISTINCT vendorid)
FROM product;
```

Query 15:
```
SELECT COUNT(*), AVG(productprice), MIN(productprice), MAX(productprice)
FROM product
WHERE categoryid = 'CP';
```

Query 16:
```
SELECT vendorid, COUNT(*), AVG(productprice)
FROM product
GROUP BY vendorid;
```

Query 17:
```
SELECT COUNT(*), AVG(productprice)
FROM product
GROUP BY vendorid;
```

Query 18:
```
SELECT vendorid, COUNT(*)
FROM product
WHERE productprice >= 100
GROUP BY vendorid;
```

Query 19:
```
SELECT vendorid, categoryid, COUNT(*), AVG(productprice)
FROM product
GROUP BY vendorid, categoryid;
```

Query 20:
```
SELECT productid, SUM(noofitems)
FROM soldvia
GROUP BY productid;
```

Query 21:
```
SELECT productid, COUNT(tid)
FROM soldvia
GROUP BY productid;
```

Query 22:
```
SELECT vendorid, categoryid, COUNT(*), AVG(productprice)
FROM product
GROUP BY vendorid, categoryid
HAVING COUNT(*) > 1;
```

Query 23:
```
SELECT vendorid, categoryid, COUNT(*), AVG(productprice)
FROM product
WHERE productprice >= 50
GROUP BY vendorid, categoryid
HAVING COUNT(*) > 1;
```
Query 24:
SELECT productid, SUM(noofitems)
FROM soldvia
GROUP BY productid
HAVING SUM(noofitems) > 3;

Query 25:
SELECT productid, COUNT(tid)
FROM soldvia
GROUP BY productid
HAVING COUNT(tid) > 1;

Query 26:
SELECT productid
FROM soldvia
GROUP BY productid
HAVING SUM(noofitems) > 3;

Query 27:
SELECT productid
FROM soldvia
GROUP BY productid
HAVING COUNT(tid) > 1;

Query 28:
SELECT productid, productname, productprice
FROM product
WHERE productprice < (SELECT AVG(productprice)
FROM product);

Query 29:
SELECT productid, productname, productprice
FROM product
WHERE productid IN (SELECT productid
FROM soldvia
GROUP BY productid
HAVING SUM(noofitems) > 3);

Query 30:
SELECT productid, productname, productprice
FROM product
WHERE productid IN (SELECT productid
FROM soldvia
GROUP BY productid
HAVING COUNT(tid) > 1);

Query 31:
SELECT productid, productname, vendorname, productprice
FROM product, vendor
WHERE product.vendorid = vendor.vendorid;

Query 32:
SELECT productid, productname, vendorname, productprice
FROM product, vendor;

Query 33:
SELECT *
FROM product, vendor;

Query 34:
SELECT *
FROM product, vendor
WHERE product.vendorid = vendor.vendorid;
Query 31a:

```
SELECT p.productid, p.productname, v.vendorname,
p.productprice
FROM product p, vendor v
WHERE p.vendorid = v.vendorid;
```

Query 31b:

```
SELECT p.productid pid, p.productname pname,
v.vendorname vname, p.productprice pprice
FROM product p, vendor v
WHERE p.vendorid = v.vendorid;
```

Query 31c:

```
SELECT p.productid AS pid, p.productname AS pname,
v.vendorname AS vname, p.productprice AS pprice
FROM product p, vendor v
WHERE p.vendorid = v.vendorid;
```

Query 35:

```
SELECT t.tid, t.tdate, p.productname, sv.noofitems AS quantity, (sv.noofitems * p.productprice) AS amount
FROM product p, salestransaction t, soldvia sv
WHERE sv.productid = p.productid AND sv.tid = t.tid
ORDER BY t.tid;
```

Alter Statement 1:

```
ALTER TABLE vendor ADD vendorphonenumber CHAR(11);
```

Alter Statement 2:

```
ALTER TABLE vendor DROP COLUMN vendorphonenumber;
```

Insert Statement 1:

```
INSERT INTO product VALUES ('7X7','Airy Sock',1000,'MK','CP');
```

Update Statement 1:

```
UPDATE product
SET productprice = 10
WHERE productid = '7X7';
```

Alter Statement 3:

```
ALTER TABLE product ADD discount CHAR(11) ;
```

Update Statement 2:

```
UPDATE product
SET discount = 0.2;
```

Update Statement 3:

```
UPDATE product
SET discount = 0.3
WHERE vendorid = 'MK';
```

Alter Statement 4:

```
ALTER TABLE product DROP COLUMN discount;
```

Delete Statement 1:

```
DELETE FROM product
WHERE productid = '7X7';
```
Create View Statement 1:
CREATE VIEW products_more_than_3_sold AS
SELECT productid, productname, productprice
FROM product
WHERE productid IN (SELECT productid
FROM soldvia
GROUP BY productid
HAVING SUM(noofitems) > 3);

Query 29a:
SELECT *
FROM products_more_than_3_sold;

Create View Statement 2:
CREATE VIEW products_in_multiple_trnsc AS
SELECT productid, productname, productprice
FROM product
WHERE productid IN (SELECT productid
FROM soldvia
GROUP BY productid
HAVING COUNT(tid) > 1);

Query 30a:
SELECT *
FROM products_in_multiple_trnsc;

Drop View Statement 1:
DROP VIEW products_more_than_3_sold;

Drop View Statement 2:
DROP VIEW products_in_multiple_trnsc;

Query 36:
SELECT *
FROM products_more_than_3_sold
UNION
SELECT *
FROM products_in_multiple_trnsc;

Query 37:
SELECT *
FROM products_more_than_3_sold
INTERSECT
SELECT *
FROM products_in_multiple_trnsc;

Query 38:
SELECT *
FROM products_more_than_3_sold
EXCEPT
SELECT *
FROM products_in_multiple_trnsc;
CREATE TABLE manager
(
    managerid CHAR(4) NOT NULL,
    mfname VARCHAR(15) NOT NULL,
    mlname VARCHAR(15) NOT NULL,
    mbdate DATE NOT NULL,
    msalary NUMERIC (9,2) NOT NULL,
    mbonus NUMERIC (9,2),
    mresbuildingid CHAR(3),
    PRIMARY KEY (managerid)
);

CREATE TABLE managerphone
(
    managerid CHAR(4) NOT NULL,
    mphone CHAR(11) NOT NULL,
    PRIMARY KEY (managerid, mphone),
    FOREIGN KEY (managerid) REFERENCES manager(managerid)
);

CREATE TABLE building
(
    buildingid CHAR(3) NOT NULL,
    bnooffloors INT NOT NULL,
    bmanagerid CHAR(4) NOT NULL,
    PRIMARY KEY (buildingid),
    FOREIGN KEY (bmanagerid) REFERENCES manager(managerid)
);

CREATE TABLE inspector
(
    insid CHAR(3) NOT NULL,
    insname VARCHAR(15) NOT NULL,
    PRIMARY KEY (insid)
);

CREATE TABLE inspecting
(
    insid CHAR(3) NOT NULL,
    buildingid CHAR(3) NOT NULL,
    datelast DATE NOT NULL,
    datenext DATE NOT NULL,
    PRIMARY KEY (insid, buildingid),
    FOREIGN KEY (insid) REFERENCES inspector(insid),
    FOREIGN KEY (buildingid) REFERENCES building(buildingid)
);

CREATE TABLE corpclient
(
    ccid CHAR(4) NOT NULL,
    ccname VARCHAR(25) NOT NULL,
    ccindustry VARCHAR(25) NOT NULL,
    cclocation VARCHAR(25) NOT NULL,
    ccidreferredby CHAR(4),
    PRIMARY KEY (ccid),
    UNIQUE (ccname),
    FOREIGN KEY (ccidreferredby) REFERENCES corpclient(ccid)
);
CREATE TABLE apartment
(
  buildingid  CHAR(3)  NOT NULL,
  aptno      CHAR(5)  NOT NULL,
  anoofbedrooms  INT  NOT NULL,
  ccid     CHAR(4),
  PRIMARY KEY (buildingid, aptno),
  FOREIGN KEY (buildingid) REFERENCES building(buildingid),
  FOREIGN KEY (ccid) REFERENCES corpclient(ccid) );

CREATE TABLE staffmember
(
  smemberid     CHAR(4)  NOT NULL,
  smembername  VARCHAR(15)  NOT NULL,
  PRIMARY KEY (smemberid) );

CREATE TABLE cleaning
(
  buildingid     CHAR(3)  NOT NULL,
  aptno      CHAR(5)  NOT NULL,
  smemberid     CHAR(4)  NOT NULL,
  CONSTRAINT cleaningpk PRIMARY KEY (buildingid, aptno, smemberid ),
  CONSTRAINT cleaningfk FOREIGN KEY (buildingid, aptno)
REFERENCES apartment(buildingid, aptno) );

INSERT INTO manager VALUES ('M12', 'Boris', 'Grant', '20/Jun/1980', 60000, null, null);
INSERT INTO manager VALUES ('M23', 'Austin', 'Lee', '30/Oct/1975', 50000, 5000, null);
INSERT INTO manager VALUES ('M34', 'George', 'Sherman', '11/Jan/1976', 52000, 2000, null);
INSERT INTO managerphone VALUES ('M12', '555-2222');
INSERT INTO managerphone VALUES ('M12', '555-3232');
INSERT INTO managerphone VALUES ('M23', '555-9988');
INSERT INTO managerphone VALUES ('M34', '555-9999');
INSERT INTO building VALUES ('B1', '5', 'M12');
INSERT INTO building VALUES ('B2', '6', 'M23');
INSERT INTO building VALUES ('B3', '4', 'M23');
INSERT INTO building VALUES ('B4', '4', 'M34');
INSERT INTO inspector VALUES ('I11', 'Jane');
INSERT INTO inspector VALUES ('I22', 'Niko');
INSERT INTO inspector VALUES ('I33', 'Mick');
INSERT INTO inspecting VALUES ('I11','B1','15/May/2012','14/May/2013');
INSERT INTO inspecting VALUES ('I11','B2','17/Feb/2013','17/May/2013');
INSERT INTO inspecting VALUES ('I22','B2','17/Feb/2013','17/May/2013');
INSERT INTO inspecting VALUES ('I22','B3','11/Jan/2013','11/Jan/2014');
INSERT INTO inspecting VALUES ('I33','B3','12/Jan/2013','12/Jan/2014');
INSERT INTO inspecting VALUES ('I33','B4','11/Jan/2013','11/Jan/2014');
ALTER TABLE manager
ADD CONSTRAINT fkresidesin
FOREIGN KEY (mresbuildingid) REFERENCES building(buildingid);

UPDATE manager
SET mresbuildingid = 'B1'
WHERE managerid = 'M12';

UPDATE manager
SET mresbuildingid = 'B2'
WHERE managerid = 'M23';

UPDATE manager
SET mresbuildingid = 'B4'
WHERE managerid = 'M34';

ALTER TABLE manager
ALTER COLUMN mresbuildingid CHAR(3) NOT NULL;

DROP TABLE cleaning;
DROP TABLE staffmember;
DROP TABLE apartment;
DROP TABLE corpclient;
DROP TABLE inspecting;
DROP TABLE inspector;
DROP TABLE managerphone;

INSERT INTO corpclient VALUES ('C111', 'BlingNotes', 'Music', 'Chicago', null);
INSERT INTO corpclient VALUES ('C222', 'SkyJet', 'Airline', 'Oak Park', 'C111');
INSERT INTO corpclient VALUES ('C777', 'WindyCT', 'Music', 'Chicago', 'C222');
INSERT INTO corpclient VALUES ('C888', 'SouthAlps', 'Sports', 'Rosemont', 'C777');

INSERT INTO apartment VALUES ('B1', '21', 1, 'C111');
INSERT INTO apartment VALUES ('B1', '41', 1, null);
INSERT INTO apartment VALUES ('B2', '11', 2, 'C222');
INSERT INTO apartment VALUES ('B2', '31', 2, null);
INSERT INTO apartment VALUES ('B3', '11', 2, 'C777');
INSERT INTO apartment VALUES ('B4', '11', 2, 'C777');

INSERT INTO staffmember VALUES ('5432', 'Brian');
INSERT INTO staffmember VALUES ('9876', 'Boris');
INSERT INTO staffmember VALUES ('7652', 'Caroline');

INSERT INTO cleaning VALUES ('B1', '21', '5432');
INSERT INTO cleaning VALUES ('B1', '41', '9876');
INSERT INTO cleaning VALUES ('B2', '31', '5432');
INSERT INTO cleaning VALUES ('B2', '11', '9876');
INSERT INTO cleaning VALUES ('B3', '11', '5432');
INSERT INTO cleaning VALUES ('B4', '11', '7652');

ALTER TABLE manager
ADD CONSTRAINT fkresidesin
FOREIGN KEY (mresbuildingid) REFERENCES building(buildingid);

UPDATE manager
SET mresbuildingid = 'B1'
WHERE managerid = 'M12';

UPDATE manager
SET mresbuildingid = 'B2'
WHERE managerid = 'M23';

UPDATE manager
SET mresbuildingid = 'B4'
WHERE managerid = 'M34';

ALTER TABLE manager
ALTER COLUMN mresbuildingid CHAR(3) NOT NULL;

DROP TABLE cleaning;
DROP TABLE staffmember;
DROP TABLE apartment;
DROP TABLE corpclient;
DROP TABLE inspecting;
DROP TABLE inspector;
DROP TABLE managerphone;
Alter Statement 7:
ALTER TABLE manager
DROP CONSTRAINT fkresidesin;

Drop Table Sequence
Database – Last Two Tables (a) & (b):
DROP TABLE building;
DROP TABLE manager;
DROP TABLE manager;
DROP TABLE building;

Query 39:
SELECT c.ccname AS client, r.ccname AS recommender
FROM corpclient c, corpclient r
WHERE r.ccid = c.ccidreferredby;

Query 40:
SELECT a.buildingid, a.aptno, c.ccname
FROM apartment a, corpclient c
WHERE a.ccid = c.ccid;

Query 41:
SELECT a.buildingid, a.aptno, c.ccname
FROM apartment a LEFT OUTER JOIN corpclient c
ON a.ccid = c.ccid;

Query 42:
SELECT a.buildingid, a.aptno, c.ccname
FROM apartment a RIGHT OUTER JOIN corpclient c
ON a.ccid = c.ccid;

Query 43:
SELECT a.buildingid, a.aptno, c.ccname
FROM apartment a FULL OUTER JOIN corpclient c
ON a.ccid = c.ccid;

Query 44:
SELECT m.managerid, m.mfname, m.mlname, s.smemberid
FROM manager m, staffmember s
WHERE m.mfname = s.smembername;

Query 45:
SELECT *
FROM manager
WHERE mbonus IS NULL;

Query 46:
SELECT *
FROM building
WHERE EXISTS
(SELECT *
FROM manager
WHERE buildingid = mresbuildingid);

Query 47:
SELECT *
FROM building
WHERE NOT EXISTS
(SELECT *
FROM manager
WHERE buildingid = mresbuildingid);
CREATE TABLE cleaningdenormalized
{
    buildingid    CHAR(3)    NOT NULL,
    aptno        CHAR(5)    NOT NULL,
    smemberid    CHAR(4)    NOT NULL,
    smembername  VARCHAR(15)    NOT NULL,
    PRIMARY KEY (buildingid, aptno, smemberid));

INSERT INTO cleaningdenormalized
SELECT  c.buildingid, c.aptno, s.smemberid, s.smembername
FROM  cleaning c, staffmember s
WHERE  c.smemberid = s.smemberid;