**7. SQL Output Printout**

SQL> SET ECHO ON

SQL> /\*

SQL> CIS 353 -Database Design Project

SQL> Benjamin Summers

SQL> Matthew Shrider

SQL> Andrew Sheffer

SQL> James Uhe

SQL> \*/

SQL> --

SQL> CREATE TABLE Supplier(

2 sname CHAR(30) PRIMARY KEY,

3 sphone VARCHAR(14),

4 saddress VARCHAR(50)

5 );

Table created.

SQL> CREATE TABLE Owner(

2 ownerid INTEGER PRIMARY KEY,

3 oname CHAR(30),

4 oaddress VARCHAR(50)

5 );

Table created.

SQL> CREATE TABLE Employee(

2 eid INTEGER PRIMARY KEY,

3 name CHAR(30),

4 address VARCHAR(50),

5 ehours INTEGER,

6 managerid INTEGER,

7 sid INTEGER ,

8 CONSTRAINT employeeic1 FOREIGN KEY (managerid) REFERENCES Employee(eid)

9 ON DELETE SET NULL,

10 CONSTRAINT employeeic2 CHECK (managerid != eid),

11 CONSTRAINT employeeic3 CHECK (ehours > 0)

12 --employeeic4 is a trigger implemented at the end of constraints.

13 --

14 --END EMPLOYEE CONSTRAINTS

15 );

Table created.

SQL> CREATE TABLE Item(

2 ibarcode INTEGER PRIMARY KEY,

3 ibrand CHAR(30),

4 iprice INTEGER,

5 iname CHAR(30),

6 sname CHAR(30),

7 --iIC1 An Item must cost more than 0

8 CONSTRAINT itemic1 CHECK (iprice>0),

9 --iIC2 the barcode of an item must be greated than zero

10 CONSTRAINT itemic2 CHECK (ibarcode>0),

11 --iIC3 Every item must have a supplier name in the Supplier table

12 CONSTRAINT itemic3 FOREIGN KEY (sname) REFERENCES Supplier(sname)

13 ON DELETE SET NULL

14 --

15 --END ITEM CONSTRAINTS

16 );

Table created.

SQL> CREATE TABLE Store(

2 sid INTEGER PRIMARY KEY,

3 slocation VARCHAR(50),

4 ownerid INTEGER NOT NULL,

5 --sIC1 osnerid is a foreign key referenced from owner

6 CONSTRAINT storeic1 FOREIGN KEY (ownerid) REFERENCES Owner(ownerid),

7 CONSTRAINT storeic2 UNIQUE(ownerid)

8 --

9 --END STORE CONSTRAINTS

10 );

Table created.

SQL> CREATE TABLE Aisle(

2 rnum INTEGER,

3 sid INTEGER,

4 islecategory CHAR(30),

5 timestocked INTEGER,

6 CONSTRAINT aisleKey PRIMARY KEY (rnum, sid),

7 --aIC2: sid must be a store id

8 CONSTRAINT aisleic2 FOREIGN KEY (sid) REFERENCES Store (sid)

9 ON DELETE CASCADE,

10 --stockedByic3 timestocked must be in correct clock format, between 0 and 24 hours

11 CONSTRAINT aisleic3 CHECK (timestocked >= 0 AND timestocked <= 2400)

12 --

13 --END AISLE CONSTRAINTS

14 );

Table created.

SQL> CREATE TABLE Shipment(

2 sname CHAR(30),

3 sid INTEGER,

4 dateshipped VARCHAR(10),

5 --sIC1 sname sid and dateshipped are the keys of this table

6 CONSTRAINT shipmentKey PRIMARY KEY (sname, sid, dateshipped),

7 --sIC2: sid must be a store id

8 CONSTRAINT shipmentic2 FOREIGN KEY (sid) REFERENCES Store (sid)

9 ON DELETE CASCADE,

10 --shipmentIC3: sname must be a suppliers name

11 CONSTRAINT shipmentic3 FOREIGN KEY (sname) REFERENCES Supplier(sname)

12 ON DELETE CASCADE

13 --

14 --END SHIPMENT CONSTRAINTS

15 );

Table created.

SQL> CREATE TABLE ItemsPerAisle(

2 ibarcode INTEGER,

3 rnum INTEGER,

4 sid INTEGER,

5 --itemsPerAisleKey ibarcode, rnum and sid are the keys

6 CONSTRAINT itemsPerAisleKey PRIMARY KEY (ibarcode, rnum, sid),

7 --itemsPerAisleic2 ibarcode must be a barcode in Item

8 CONSTRAINT itemsPerAisleic2 FOREIGN KEY (ibarcode) REFERENCES Item(ibarcode),

9 --itemsPerAisleic3: sid must be a store id

10 CONSTRAINT itemsPerAisleic3 FOREIGN KEY (rnum,sid) REFERENCES Aisle (rnum,sid)

11 ON DELETE CASCADE

12 --

13 --END ITEMSPERAISLE CONSTRAINTS

14 );

Table created.

SQL> CREATE TABLE StockedBy(

2 eid INTEGER NOT NULL,

3 rnum INTEGER NOT NULL,

4 sid INTEGER NOT NULL,

5 --stockedByKey eid, rnum, and sid are the keys

6 CONSTRAINT stockedByic1 PRIMARY KEY (eid, rnum, sid),

7 --stockedByic4 rnum is a foreign key from aisle rnum

8 CONSTRAINT stockedByic3 FOREIGN KEY (rnum,sid) REFERENCES Aisle(rnum,sid)

9 ON DELETE CASCADE,

10 --stockedByic5 eid is a foreign key from employee

11 CONSTRAINT stockedByic4 FOREIGN KEY (eid) REFERENCES Employee(eid)

12 ON DELETE CASCADE

13 --

14 --END STOCKEDBY CONSTRAINTS

15 );

Table created.

SQL> CREATE TABLE EmployeePhone(

2 eid INTEGER,

3 ephone VARCHAR(14),

4 --employeePhoneKey eid and ephone are keys

5 CONSTRAINT employeePhoneKey PRIMARY KEY (eid, ephone),

6 --employeePhoneic1 eid is a foreign key referencing eid from employee

7 CONSTRAINT employeePhoneic1 FOREIGN KEY (eid) REFERENCES Employee(eid)

8 );

Table created.

SQL> --Trigger for employeeIC5: every employee must work at the same

SQL> --store as their manager.

SQL> CREATE OR REPLACE TRIGGER employeeIC5

2 AFTER INSERT OR UPDATE OF sid ON Employee

3 DECLARE numFound INTEGER;

4 BEGIN

5 SELECT COUNT(\*)

6 INTO numFound

7 FROM EMPLOYEE E, EMPLOYEE E2

8 WHERE E.managerid = E2.eid AND E.sid != E2.sid;

9 IF numFound > 0

10 THEN

11 RAISE\_APPLICATION\_ERROR(-20001, 'There is at least one manager

12 and employee who do not work at the same

13 store.');

14 END IF;

15 END;

16 /

Trigger created.

SQL> --Trigger for employeeIC4: every employee must work at the same

SQL> --store as their manager.

SQL> CREATE OR REPLACE TRIGGER employeeIC4

2 AFTER INSERT OR UPDATE OF managerid ON Employee

3 DECLARE numFound INTEGER;

4 BEGIN

5 SELECT COUNT(\*)

6 INTO numFound

7 FROM EMPLOYEE E, EMPLOYEE E2

8 WHERE E.managerid = E2.eid AND E.sid != E2.sid;

9 IF numFound > 0

10 THEN

11 RAISE\_APPLICATION\_ERROR(-20001, 'There is at least one manager

12 and employee who do not work at the same

13 store.');

14 END IF;

15 END;

16 /

Trigger created.

SQL> --

SQL> --END Schema

SQL> --

SQL> --< The INSERT statements that populate the tables>

SQL> ALTER TRIGGER employeeIC4 DISABLE;

Trigger altered.

SQL> INSERT INTO Employee VALUES (147788,'Romney','10284 Ascendant Way',28,NULL,244);

1 row created.

SQL> INSERT INTO Employee VALUES (138888,'Lindsy','6472 Woodward Rd',15,NULL,233);

1 row created.

SQL> INSERT INTO Employee VALUES (123333,'Peter','0001 Patard Street',40,NULL,222);

1 row created.

SQL> INSERT INTO Employee VALUES (121111,'John','2998 Nobody Street',20,123333,222);

1 row created.

SQL> INSERT INTO Employee VALUES (122222,'Susie','1090 Whovill Ave',25,123333,222);

1 row created.

SQL> INSERT INTO Employee VALUES (123565,'Nikky','4989 Helpers Dr',22,123333,222);

1 row created.

SQL> INSERT INTO Employee VALUES (135454,'Hulio','253 Packard Way',20,138888,233);

1 row created.

SQL> INSERT INTO Employee VALUES (135945,'Funniers','468 Joker Way',26,138888,233);

1 row created.

SQL> INSERT INTO Employee VALUES (139999,'Dave','414 Wall St',30,138888,233);

1 row created.

SQL> INSERT INTO Employee VALUES (141122,'Hue','1919 MarryMan PKWY',5,147788,244);

1 row created.

SQL> INSERT INTO Employee VALUES (145566,'Veto','1616 CoffeeMee Dr',40,147788,244);

1 row created.

SQL> INSERT INTO Employee VALUES (146582,'River','101 Arial Rd',16,147788,244);

1 row created.

SQL> ALTER TRIGGER employeeIC4 ENABLE;

Trigger altered.

SQL> --End Employee

SQL> INSERT INTO Owner VALUES (323,'Richard','Sir James AVE');

1 row created.

SQL> INSERT INTO Owner VALUES (339,'Jane','High and Mighty RD');

1 row created.

SQL> INSERT INTO Owner VALUES (345,'Beaker','Eagle DR');

1 row created.

SQL> --End Owner

SQL> INSERT INTO Store VALUES (222,'1234 No Way',323);

1 row created.

SQL> INSERT INTO Store VALUES (233,'1337 Main Street',339);

1 row created.

SQL> INSERT INTO Store VALUES (244,'3371 Walker Rd',345);

1 row created.

SQL> --End Stores

SQL> INSERT INTO Aisle VALUES (51 ,222 ,'Produce', 1200);

1 row created.

SQL> INSERT INTO Aisle VALUES (52 ,222 ,'Baked Goods',1300);

1 row created.

SQL> INSERT INTO Aisle VALUES (53 ,222 ,'Frozen Foods',1400);

1 row created.

SQL> INSERT INTO Aisle VALUES (54 ,222 ,'Cleaning',2300);

1 row created.

SQL> INSERT INTO Aisle VALUES (55 ,222 ,'Hygine',500);

1 row created.

SQL> INSERT INTO Aisle VALUES (56 ,222 ,'Pharmasuticles',900);

1 row created.

SQL> INSERT INTO Aisle VALUES (51 ,233 ,'Clothing',800);

1 row created.

SQL> INSERT INTO Aisle VALUES (52 ,233 ,'Produce',NULL);

1 row created.

SQL> INSERT INTO Aisle VALUES (53 ,233 ,'Frozen Foods',NULL);

1 row created.

SQL> INSERT INTO Aisle VALUES (54 ,233 ,'Pharmasuticles',900);

1 row created.

SQL> INSERT INTO Aisle VALUES (55 ,233 ,'Hygine',2200);

1 row created.

SQL> INSERT INTO Aisle VALUES (56 ,233 ,'Furniture',200);

1 row created.

SQL> INSERT INTO Aisle VALUES (51 ,244 ,'Furnitue', 2000);

1 row created.

SQL> INSERT INTO Aisle VALUES (52 ,244 ,'Pharmasuticles',1500);

1 row created.

SQL> INSERT INTO Aisle VALUES (53 ,244 ,'Hygine',1600);

1 row created.

SQL> INSERT INTO Aisle VALUES (54 ,244 , 'Frozen Foods',1746);

1 row created.

SQL> INSERT INTO Aisle VALUES (55 ,244 , 'BakedGoods', 1642);

1 row created.

SQL> INSERT INTO Aisle VALUES (56 ,244 ,'Produce', 1912);

1 row created.

SQL> --End Aisle

SQL> INSERT INTO EmployeePhone VALUES (121111,6165550000);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (122222,6165551122);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (123333,6165553333);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (123565,5135557777);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (135454,5135556969);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (135945,5135559999);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (138888,5662768889);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (139999,5663261578);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (141122,5662263981);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (145566,6128790338);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (146582,8107355555);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (146582,8187355555);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (147788,8108675309);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (147788,8108675308);

1 row created.

SQL> INSERT INTO EmployeePhone VALUES (147788,8108675307);

1 row created.

SQL> --End EmployeePhone

SQL> INSERT INTO StockedBy VALUES (121111, 51, 222);

1 row created.

SQL> INSERT INTO StockedBy VALUES (121111,52,222);

1 row created.

SQL> INSERT INTO StockedBy VALUES (122222,53,222);

1 row created.

SQL> INSERT INTO StockedBy VALUES (122222,54,222);

1 row created.

SQL> INSERT INTO StockedBy VALUES (122222,55,222);

1 row created.

SQL> INSERT INTO StockedBy VALUES (121111,56,222);

1 row created.

SQL> INSERT INTO StockedBy VALUES (135454,51,233);

1 row created.

SQL> INSERT INTO StockedBy VALUES (135454,52,233);

1 row created.

SQL> INSERT INTO StockedBy VALUES (135454,53,233);

1 row created.

SQL> INSERT INTO StockedBy VALUES (135945,54,233);

1 row created.

SQL> INSERT INTO StockedBy VALUES (135945,55,233);

1 row created.

SQL> INSERT INTO StockedBy VALUES (146582,51,244);

1 row created.

SQL> INSERT INTO StockedBy VALUES (146582,52,244);

1 row created.

SQL> INSERT INTO StockedBy VALUES (146582,53,244);

1 row created.

SQL> INSERT INTO StockedBy VALUES (145566,54,244);

1 row created.

SQL> INSERT INTO StockedBy VALUES (145566,55,244);

1 row created.

SQL> INSERT INTO StockedBy VALUES (145566,56,244);

1 row created.

SQL> --End StockedBy

SQL> INSERT INTO Supplier VALUES ('Live Exchange Inc' , 6155554610 ,'Tom Jones Rd');

1 row created.

SQL> INSERT INTO Supplier VALUES ('Gobi Inc' ,6155556616 , 'Nominburg Pwky');

1 row created.

SQL> INSERT INTO Supplier VALUES ('Medicine Hat Systems' ,6155546944 , 'Hogwarts Ave');

1 row created.

SQL> INSERT INTO Supplier VALUES ('Armstrong Shipments' ,6155556579 ,'Mars Dr');

1 row created.

SQL> --End Supplier

SQL> INSERT INTO Item VALUES (9211111,'Generic',4.3,'Ham Generic','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9211166,'Italiano',1.1,'Mozzarella cheese','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9211444,'Aspirin',6.3,'Aspirin','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9211555,'Dat Cereal',2.49,'Cocoa Cocoa crunchies','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9222222,'Dat Cereal',2.49,'The Munchies','Gobi Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9222345,'Bleach',12.99,'Cleaning Bleach','Gobi Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9222444,'Italiano',4.99,'Frozen Pizza','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9222555,'CHICK EM',6.35,'Chicken Pot Pie','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9245784,'Cue',0.85,'Cue Tips','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9255566,'Bleach',13.98,'Bleach Brand Bleach','Armstrong Shipments');

1 row created.

SQL> INSERT INTO Item VALUES (9377777,'Lazy Boy',112.89,'Lazy Boy Recliner','Armstrong Shipments');

1 row created.

SQL> INSERT INTO Item VALUES (9311111,'Soda Streamline',1.12,'Lemonade','Armstrong Shipments');

1 row created.

SQL> INSERT INTO Item VALUES (9322222,'Faynogo',0.89,'Grape Drink','Gobi Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9333333,'Fun Math',21.99,'Book Tarts Shirt','Gobi Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9337833,'C++',0.99,'C++ Mac and Cheese','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9344444,'Feels Gewd',0.99,'Tooth UnDecay','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9345454,'Fun Math',4.49,'Fun Math Band-aids','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9355788,'Feels Gewd',4.99,'Night-Night-Qil','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9365689,'Band-Aidtogo',19.99,'Orange-Bob Cirlce-Pants','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9388785,'Italiano',3.24,'Watermellon Beef','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9399896,'Big O',1.95,'Big ol Stick O Buttah','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9411111,'Generic',2.23,'Milk','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9411337,'CHICK EM',2.46,'Chicken of the VNC','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9422123,'Cleanly Hair',7,'Shampoo','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9422222,'Generic',4,'Donuts a Dozen','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9422636,'CHICK EM',0.99,'Big Cabbages','Gobi Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9444445,'Lazy Boy',35.99,'Wooden Stool','Gobi Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9455568,'Pointless Inc',12.99,'Giant Marker Bench','Gobi Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9455678,'Fun Math',1.99,'SQL Noodles','Live Exchange Inc');

1 row created.

SQL> INSERT INTO Item VALUES (9477778,'Hair togo',6.99,'Big Fake Hair','Armstrong Shipments');

1 row created.

SQL> INSERT INTO Item VALUES (9478745,'Feels Gewd',5.49,'High By Cough Syrup','Medicine Hat Systems');

1 row created.

SQL> INSERT INTO Item VALUES (9488785,'Big O',29.99,'Big Ol Moon Flag','Armstrong Shipments');

1 row created.

SQL> INSERT INTO Item VALUES (9499955,'Generic',6.55,'Apple Wood Smoked Bacon','Live Exchange Inc');

1 row created.

SQL> --End Item

SQL> INSERT INTO ItemsPerAisle VALUES (9211111, 51, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9211166, 51, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9211444, 56, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9377777, 52, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9222222, 52, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9222345, 54, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9222444, 53, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9222555, 53, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9245784, 55, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9255566, 55, 222);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9377777, 56, 233);

1 row created.

SQL> insert into itemsperaisle values (9211111, 52, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9322222, 52, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9333333, 51, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9337833, 52, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9344444, 55, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9345454, 54, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9355788, 54, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9365689, 54, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9388785, 53, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9399896, 52, 233);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9211111, 52, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9411337, 54, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9422123, 53, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9422222, 55, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9422636, 56, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9444445, 51, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9455568, 51, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9455678, 55, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9377777, 53, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9478745, 52, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9488785, 54, 244);

1 row created.

SQL> INSERT INTO ItemsPerAisle VALUES (9499955, 54, 244);

1 row created.

SQL> --End ItemsPerAisle

SQL> INSERT INTO Shipment VALUES ('Live Exchange Inc', 222 ,'4/10/2012');

1 row created.

SQL> INSERT INTO Shipment VALUES ('Gobi Inc' ,233 ,'4/8/2012');

1 row created.

SQL> INSERT INTO Shipment VALUES ('Medicine Hat Systems' ,233 ,'3/24/2012');

1 row created.

SQL> INSERT INTO Shipment VALUES ('Live Exchange Inc' ,222 , '12/22/2011');

1 row created.

SQL> INSERT INTO Shipment VALUES ('Medicine Hat Systems' ,244 ,'1/5/2012');

1 row created.

SQL> --End Shipment

SQL> --

SQL> COMMIT

2 --

3 --QUERY EMPLOYEE

4 SELECT \* FROM EMPLOYEE;

SELECT \* FROM EMPLOYEE

\*

ERROR at line 4:

ORA-02185: a token other than WORK follows COMMIT

SQL> --

SQL> --QUERY TABLE ITEM

SQL> --

SQL> SELECT \* FROM ITEM;

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9211111 Generic 4

Ham Generic Live Exchange Inc

9211166 Italiano 1

Mozzarella cheese Medicine Hat Systems

9211444 Aspirin 6

Aspirin Medicine Hat Systems

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9211555 Dat Cereal 2

Cocoa Cocoa crunchies Medicine Hat Systems

9222222 Dat Cereal 2

The Munchies Gobi Inc

9222345 Bleach 13

Cleaning Bleach Gobi Inc

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9222444 Italiano 5

Frozen Pizza Medicine Hat Systems

9222555 CHICK EM 6

Chicken Pot Pie Medicine Hat Systems

9245784 Cue 1

Cue Tips Live Exchange Inc

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9255566 Bleach 14

Bleach Brand Bleach Armstrong Shipments

9377777 Lazy Boy 113

Lazy Boy Recliner Armstrong Shipments

9311111 Soda Streamline 1

Lemonade Armstrong Shipments

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9322222 Faynogo 1

Grape Drink Gobi Inc

9333333 Fun Math 22

Book Tarts Shirt Gobi Inc

9337833 C++ 1

C++ Mac and Cheese Live Exchange Inc

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9344444 Feels Gewd 1

Tooth UnDecay Live Exchange Inc

9345454 Fun Math 4

Fun Math Band-aids Medicine Hat Systems

9355788 Feels Gewd 5

Night-Night-Qil Medicine Hat Systems

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9365689 Band-Aidtogo 20

Orange-Bob Cirlce-Pants Medicine Hat Systems

9388785 Italiano 3

Watermellon Beef Live Exchange Inc

9399896 Big O 2

Big ol Stick O Buttah Live Exchange Inc

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9411111 Generic 2

Milk Live Exchange Inc

9411337 CHICK EM 2

Chicken of the VNC Live Exchange Inc

9422123 Cleanly Hair 7

Shampoo Medicine Hat Systems

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9422222 Generic 4

Donuts a Dozen Live Exchange Inc

9422636 CHICK EM 1

Big Cabbages Gobi Inc

9444445 Lazy Boy 36

Wooden Stool Gobi Inc

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9455568 Pointless Inc 13

Giant Marker Bench Gobi Inc

9455678 Fun Math 2

SQL Noodles Live Exchange Inc

9477778 Hair togo 7

Big Fake Hair Armstrong Shipments

IBARCODE IBRAND IPRICE

---------- ------------------------------ ----------

INAME SNAME

------------------------------ ------------------------------

9478745 Feels Gewd 5

High By Cough Syrup Medicine Hat Systems

9488785 Big O 30

Big Ol Moon Flag Armstrong Shipments

9499955 Generic 7

Apple Wood Smoked Bacon Live Exchange Inc

33 rows selected.

SQL> --

SQL> --QUERY TABLE SHIPMENT

SQL> --

SQL> SELECT \* FROM SHIPMENT;

SNAME SID DATESHIPPE

------------------------------ ---------- ----------

Gobi Inc 233 4/8/2012

Live Exchange Inc 222 12/22/2011

Live Exchange Inc 222 4/10/2012

Medicine Hat Systems 233 3/24/2012

Medicine Hat Systems 244 1/5/2012

SQL> --

SQL> --QUERY TABLE STORE

SQL> --

SQL> SELECT \* FROM STORE;

SID SLOCATION OWNERID

---------- -------------------------------------------------- ----------

222 1234 No Way 323

233 1337 Main Street 339

244 3371 Walker Rd 345

SQL> --

SQL> --QUERY TABLE AISLE;

SQL> --

SQL> SELECT \* FROM AISLE;

RNUM SID ISLECATEGORY TIMESTOCKED

---------- ---------- ------------------------------ -----------

51 222 Produce 1200

52 222 Baked Goods 1300

53 222 Frozen Foods 1400

54 222 Cleaning 2300

55 222 Hygine 500

56 222 Pharmasuticles 900

51 233 Clothing 800

52 233 Produce

53 233 Frozen Foods

54 233 Pharmasuticles 900

55 233 Hygine 2200

RNUM SID ISLECATEGORY TIMESTOCKED

---------- ---------- ------------------------------ -----------

56 233 Furniture 200

51 244 Furnitue 2000

52 244 Pharmasuticles 1500

53 244 Hygine 1600

54 244 Frozen Foods 1746

55 244 BakedGoods 1642

56 244 Produce 1912

18 rows selected.

SQL> --

SQL> --QUERY TABLE STOCKEDBY

SQL> --

SQL> SELECT \* FROM STOCKEDBY;

EID RNUM SID

---------- ---------- ----------

121111 51 222

121111 52 222

121111 56 222

122222 53 222

122222 54 222

122222 55 222

135454 51 233

135454 52 233

135454 53 233

135945 54 233

135945 55 233

EID RNUM SID

---------- ---------- ----------

145566 54 244

145566 55 244

145566 56 244

146582 51 244

146582 52 244

146582 53 244

17 rows selected.

SQL> --

SQL> --QUERY TABLE EMPLOYEEPHONE

SQL> --

SQL> SELECT \* FROM EMPLOYEEPHONE;

EID EPHONE

---------- --------------

121111 6165550000

122222 6165551122

123333 6165553333

123565 5135557777

135454 5135556969

135945 5135559999

138888 5662768889

139999 5663261578

141122 5662263981

145566 6128790338

146582 8107355555

EID EPHONE

---------- --------------

146582 8187355555

147788 8108675307

147788 8108675308

147788 8108675309

15 rows selected.

SQL> --

SQL> --QUERY TABLE ITEMSPERAISLE

SQL> --

SQL> SELECT \* FROM ITEMSPERAISLE;

IBARCODE RNUM SID

---------- ---------- ----------

9211111 51 222

9211111 52 233

9211111 52 244

9211166 51 222

9211444 56 222

9222222 52 222

9222345 54 222

9222444 53 222

9222555 53 222

9245784 55 222

9255566 55 222

IBARCODE RNUM SID

---------- ---------- ----------

9322222 52 233

9333333 51 233

9337833 52 233

9344444 55 233

9345454 54 233

9355788 54 233

9365689 54 233

9377777 52 222

9377777 53 244

9377777 56 233

9388785 53 233

IBARCODE RNUM SID

---------- ---------- ----------

9399896 52 233

9411337 54 244

9422123 53 244

9422222 55 244

9422636 56 244

9444445 51 244

9455568 51 244

9455678 55 244

9478745 52 244

9488785 54 244

9499955 54 244

33 rows selected.

SQL> --

SQL> --QUERY TABLE SUPPLIER

SQL> --

SQL> SELECT \* FROM SUPPLIER;

SNAME SPHONE

------------------------------ --------------

SADDRESS

--------------------------------------------------

Live Exchange Inc 6155554610

Tom Jones Rd

Gobi Inc 6155556616

Nominburg Pwky

Medicine Hat Systems 6155546944

Hogwarts Ave

SNAME SPHONE

------------------------------ --------------

SADDRESS

--------------------------------------------------

Armstrong Shipments 6155556579

Mars Dr

SQL> --

SQL> --QUERY TABLE OWNER

SQL> SELECT \* FROM OWNER;

OWNERID ONAME

---------- ------------------------------

OADDRESS

--------------------------------------------------

323 Richard

Sir James AVE

339 Jane

High and Mighty RD

345 Beaker

Eagle DR

SQL> --

SQL> --

SQL> --QUERY 1:join involving four relations

SQL> -- Every employee who has a manager and works for a store whose owner id

SQL> -- is 323.

SQL> SELECT E.name, P.ephone, E.managerid, S.slocation , O.oname

2 FROM EMPLOYEE E, EMPLOYEEPHONE P, STORE S, Owner O

3 WHERE E.eid = P.eid AND E.managerid IS NOT NULL AND

4 E.sid=S.sid AND S.ownerid = O.ownerid AND O.ownerid = 323

5 ORDER BY S.slocation;

NAME EPHONE MANAGERID

------------------------------ -------------- ----------

SLOCATION

--------------------------------------------------

ONAME

------------------------------

John 6165550000 123333

1234 No Way

Richard

Susie 6165551122 123333

1234 No Way

Richard

NAME EPHONE MANAGERID

------------------------------ -------------- ----------

SLOCATION

--------------------------------------------------

ONAME

------------------------------

Nikky 5135557777 123333

1234 No Way

Richard

SQL> --

SQL> --QUERY 2: self-join

SQL> -- Pairs of employees that work the same amount of hours.

SQL> SELECT E1.name, E2.name, E2.ehours

2 FROM EMPLOYEE E1, EMPLOYEE E2

3 WHERE E1.ehours=E2.ehours AND E1.eid > E2.eid;

NAME NAME EHOURS

------------------------------ ------------------------------ ----------

Veto Peter 40

Hulio John 20

SQL> ---

SQL> --QUERY 3: MINUS

SQL> --Find every item that cost more than 10 dollars and is not supplied by

SQL> --Gobi Inc.

SQL> SELECT I.ibarcode, I.iname, I.iprice

2 FROM ITEM I

3 WHERE I.iprice > 10

4 MINUS

5 SELECT I.ibarcode, I.iname, I.iprice

6 FROM ITEM I

7 WHERE I.sname= 'Gobi Inc';

IBARCODE INAME IPRICE

---------- ------------------------------ ----------

9255566 Bleach Brand Bleach 14

9365689 Orange-Bob Cirlce-Pants 20

9377777 Lazy Boy Recliner 113

9488785 Big Ol Moon Flag 30

SQL> ----

SQL> --QUERY 4: AVG()

SQL> --Find the average hours of each employee's store

SQL> SELECT E.sid, AVG(ehours)

2 FROM EMPLOYEE E

3 Group By E.sid

4 ORDER BY AVG(ehours);

SID AVG(EHOURS)

---------- -----------

244 22.25

233 22.75

222 26.75

SQL> --QUERY 5:GROUP BY,HAVING,ORDER BY

SQL> --For each store list the number of itmes in each aisle when the aisle contains

SQL> --more than 2 items.

SQL> SELECT I.sid, I.rnum, COUNT(\*) AS NumItems

2 FROM ITEMSPERAISLE I

3 GROUP BY I.sid, I.rnum

4 HAVING COUNT(\*) > 2

5 ORDER BY I.rnum;

SID RNUM NUMITEMS

---------- ---------- ----------

233 52 4

233 54 3

244 54 3

SQL> --

SQL> --QUERY 7: non-correlated subquery

SQL> --Find all employees who work more than 10 hours and have not stocked aisle's

SQL> SELECT E.eid, E.name

2 FROM EMPLOYEE E

3 WHERE E.ehours > 10 AND

4 E.eid NOT IN (SELECT S1.eid

5 FROM STOCKEDBY S1

6 WHERE S1.SID !=222);

EID NAME

---------- ------------------------------

138888 Lindsy

139999 Dave

147788 Romney

121111 John

122222 Susie

123565 Nikky

123333 Peter

7 rows selected.

SQL> --

SQL> --QUERY 8: DIVISION operator and correlated subquery

SQL> --Find all the barcode and brand name of all the items that are in every store.

SQL> SELECT DISTINCT I.ibarcode, I.ibrand

2 FROM Item I, ItemsPerAisle IP

3 WHERE NOT EXISTS(( SELECT S.sid

4 FROM Store S)

5 MINUS

6 (SELECT IPA.sid

7 FROM ItemsPerAisle IPA

8 WHERE IPA.ibarcode = I.ibarcode));

IBARCODE IBRAND

---------- ------------------------------

9377777 Lazy Boy

9211111 Generic

SQL> --

SQL> --QUERY 9: Outer join

SQL> --Find every employee who has more than one phone number.

SQL> SELECT E.eid, E.name

2 FROM employeePhone EP LEFT OUTER JOIN employee E

3 ON E.eid = EP.eid

4 GROUP BY E.eid, E.name

5 HAVING COUNT(\*) >1;

EID NAME

---------- ------------------------------

146582 River

147788 Romney

SQL> --

SQL> --

SQL> -- Constratin Tests -- not completed

SQL> -- Tests Stockedby ic1 - duplicate primary keys or null values

SQL> INSERT INTO StockedBy VALUES (121111, 51, 222);

INSERT INTO StockedBy VALUES (121111, 51, 222)

\*

ERROR at line 1:

ORA-00001: unique constraint (UHEJ.STOCKEDBYIC1) violated

SQL> INSERT INTO StockedBY Values (null, 51, 222);

INSERT INTO StockedBY Values (null, 51, 222)

\*

ERROR at line 1:

ORA-01400: cannot insert NULL into ("UHEJ"."STOCKEDBY"."EID")

SQL> -- Tests Store ic 1

SQL> INSERT INTO Store VALUES (1337,'1337 Main Street',007);

INSERT INTO Store VALUES (1337,'1337 Main Street',007)

\*

ERROR at line 1:

ORA-02291: integrity constraint (UHEJ.STOREIC1) violated - parent key not found

SQL> -- TESTS Item ic 1

SQL> INSERT INTO Item VALUES (9211111,'Generic',-10,'Ham Generic','Live Exchange Inc');

INSERT INTO Item VALUES (9211111,'Generic',-10,'Ham Generic','Live Exchange Inc')

\*

ERROR at line 1:

ORA-02290: check constraint (UHEJ.ITEMIC1) violated

SQL> --This should be a success

SQL> INSERT INTO Item VALUES (9211111,'Generic',10,'Ham Generic','Live Exchange Inc');

INSERT INTO Item VALUES (9211111,'Generic',10,'Ham Generic','Live Exchange Inc')

\*

ERROR at line 1:

ORA-00001: unique constraint (UHEJ.SYS\_C00423713) violated

SQL> -- Tests Employee ic4

SQL> --This should fail

SQL> UPDATE Employee SET ManagerID=138888

2 WHERE EID=146582;

UPDATE Employee SET ManagerID=138888

\*

ERROR at line 1:

ORA-20001: There is at least one manager

and employee who do not work at the same

store.

ORA-06512: at "UHEJ.EMPLOYEEIC4", line 9

ORA-04088: error during execution of trigger 'UHEJ.EMPLOYEEIC4'

SQL> --Test that an employee can change stores.

SQL> --This should be a success

SQL> UPDATE Employee SET sid=233, managerID=138888

2 WHERE EID=146582;

1 row updated.

SQL> SPOOL OFF