Team GLASTA's *Fantastic Furniture*: Database Initialization

Team Info

Team Name:	Team GLASTA	
Project Name:	*Fantastic Furniture*	
Participants:	Timothy Gibson	tgibson1@csustan.edu
	Alexander Altman	aaltman@csustan.edu
	Schuyler Davis	sdavis20@csustan.edu

Slip Days

- Timothy Gibson is using 1 slip day and has 3 slip days remaining.
- Alexander Altman is using 1 slip day and has 3 slip days remaining.
- Schuyler Davis is using 1 slip day and has 3 slip days remaining.

Relations

```
create domain posreal as double precision
check         (value > 0.0);

create domain posint as integer
check         (value > 0);

create table Supplier(supplierID  varchar(10),
                      name_       nchar varying(50)
                                  not null,
                      phone       varchar(12),
                      address     nchar varying(100),
                      country     char(2),
                      website     nchar varying(50),
                      primary key (supplierID));

create table Designer(designerID  varchar(10),
                      name_       nchar varying(50)
                                  not null,
                      phone       varchar(12),
                      address     nchar varying(100),
                      country     char(2),
                      website     nchar varying(50),
                      designFocus nchar varying(100),
```

```
                    primary key (designerID));

create table Set_(setID         varchar(10),
                  name_         nchar varying(50)
                                not null,
                  catalogYear   numeric(4,0),
                  catalogNumber integer
                                not null,
                  style_        nchar varying(30),
                  primary key   (setID));

create table Model(modelNumber varchar(10),
                   name_       nchar varying(50)
                               not null,
                   material    nchar varying(30),
                   upholstery  nchar varying(30),
                   durability  nchar varying(30),
                   color       nchar varying(30),
                   primary key (modelNumber));

create table Item(sku         varchar(10),
                  length_     posreal, -- in inches
                  width       posreal, -- in inches
                  height      posreal, -- in inches
                  condition   nchar varying(30),
                  weightLimit posreal, -- in pounds of weight
                  primary key (sku));

create table DistributionCenter(centerID    varchar(10),
                                name_       nchar varying(50)
                                            not null,
                                phone       varchar(12),
                                address     nchar varying(100),
                                country     char(2),
                                website     nchar varying(50),
                                primary key (centerID));

create table make(supplierID  varchar(10),
                  designerID  varchar(10),
                  setID       varchar(10),
                  primary key (supplierID,
                               designerID,
                               setID),
                  foreign key (supplierID)
                               references Supplier,
```

```
                    foreign key (designerID)
                                references Designer,
                    foreign key (setID)
                                references Set_);

create table contains_(setID       varchar(10),
                       modelNumber varchar(10),
                       count_      posint,
                       primary key (setID,
                                    modelNumber),
                       foreign key (setID)
                                   references Set_,
                       foreign key (modelNumber)
                                   references Model);

create table describes(modelNumber varchar(10)
                                   not null,
                       sku         varchar(10),
                       primary key (sku),
                       foreign key (modelNumber)
                                   references Model,
                       foreign key (sku)
                                   references Item);

create table canOrderFrom(centerID    varchar(10),
                          supplierID  varchar(10),
                          leadTime    double precision, -- in days
                          primary key (centerID,
                                       supplierID),
                          foreign key (centerID)
                                      references DistributionCenter,
                          foreign key (supplierID)
                                      references Supplier,
                          check       (leadTime >= 0.0));

create table stocks(centerID    varchar(10)
                                not null,
                    sku         varchar(10),
                    primary key (sku),
                    foreign key (centerID)
                                references DistributionCenter,
                    foreign key (sku)
                                references Item);

create table Chair(sku          varchar(10),
```

```
                    numberOfLegs posint,
                    hasCushion   boolean,
                    hasArms      boolean,
                    backHeight   posreal, -- in inches
                    seatHeight   posreal, -- in inches
                    primary key  (sku),
                    foreign key  (sku)
                                 references Item);

create table Table_(sku           varchar(10),
                     numberOfLegs  posint,
                     numberOfSeats posint,
                     shape         nchar varying(30),
                     primary key   (sku),
                     foreign key   (sku)
                                   references Item);

create table Desk(sku             varchar(10),
                  angle           double precision,
                   ↪  -- in degrees, possibly negative
                  numberOfDrawers posint,
                  primary key     (sku),
                  foreign key     (sku)
                                  references Item,
                  check              (angle > -360.0
                                  and angle < 360.0));

create table Stool(sku          varchar(10),
                    numberOfLegs posint,
                    hasCushion   boolean,
                    hasSwivel    boolean,
                    primary key  (sku),
                    foreign key  (sku)
                                 references Item);

create table Cabinet(sku                  varchar(10),
                      numberOfCompartments posint,
                      capacity             nchar varying(30),
                      primary key          (sku),
                      foreign key          (sku)
                                           references Item);

create table Bedframe(sku         varchar(10),
                       size_      nchar varying(30),
```

```
                        depth_      double precision,
                         ↪  -- in inches, possibly negative
                        primary key (sku),
                        foreign key (sku)
                                    references Item);

create table features_Feature(modelNumber varchar(10),
                              description nchar varying(50),
                              count_      posint,
                              primary key (modelNumber,
                                           description),
                              foreign key (modelNumber)
                                          references Model);
```

Table Creation Statements

```
CREATE TABLE Supplier
  (
     supplierID VARCHAR(10) CHARACTER SET ASCII,
     name_      VARCHAR(50) CHARACTER SET utf8mb4 NOT NULL,
     phone      VARCHAR(12) CHARACTER SET ASCII,
     address    VARCHAR(100) CHARACTER SET utf8mb4,
     country    CHAR(2) CHARACTER SET ASCII DEFAULT 'US',
     website    VARCHAR(50) CHARACTER SET utf8mb4,
     PRIMARY KEY(supplierID),
     CHECK (country REGEXP '^[A-Z]{2}$'),
     CHECK (phone REGEXP '^[0-9]{7,12}$')
  );

CREATE TABLE Designer
  (
     designerID  VARCHAR(10) CHARACTER SET ASCII,
     name_       VARCHAR(50) CHARACTER SET utf8mb4 NOT NULL,
     phone       VARCHAR(12) CHARACTER SET ASCII,
     address     VARCHAR(100) CHARACTER SET utf8mb4,
     country     CHAR(2) CHARACTER SET ASCII DEFAULT 'US',
     website     VARCHAR(50) CHARACTER SET utf8mb4,
     designFocus VARCHAR(100) CHARACTER SET utf8mb4,
     PRIMARY KEY(designerID),
     CHECK (country REGEXP '^[A-Z]{2}$'),
     CHECK (phone REGEXP '^[0-9]{7,12}$')
  );

CREATE TABLE Set_
```

```
  (
     setID         VARCHAR(10) CHARACTER SET ASCII,
     name_         VARCHAR(50) CHARACTER SET utf8mb4 NOT NULL,
     catalogYear   DECIMAL(4, 0) UNSIGNED,
     catalogNumber BIGINT UNSIGNED ZEROFILL NOT NULL,
     style_        VARCHAR(30) CHARACTER SET utf8mb4,
     PRIMARY KEY(setID)
  );

CREATE TABLE Model
  (
     modelNumber VARCHAR(10) CHARACTER SET ASCII,
     name_       VARCHAR(50) CHARACTER SET utf8mb4 NOT NULL,
     material    VARCHAR(30) CHARACTER SET utf8mb4,
     upholstery  VARCHAR(30) CHARACTER SET utf8mb4,
     durability  VARCHAR(30) CHARACTER SET utf8mb4,
     color       VARCHAR(30) CHARACTER SET utf8mb4,
     PRIMARY KEY(modelNumber)
  );

CREATE TABLE Item
  (
     sku         VARCHAR(10) CHARACTER SET ASCII,
     length_     DOUBLE UNSIGNED,
     width       DOUBLE UNSIGNED,
     height      DOUBLE UNSIGNED,
     condition_  VARCHAR(30) CHARACTER SET utf8mb4,
     weightLimit DOUBLE UNSIGNED,
     PRIMARY KEY(sku),
     CHECK (length_ > 0.0),
     CHECK (width > 0.0),
     CHECK (height > 0.0),
     CHECK (weightLimit > 0.0)
  );

CREATE TABLE DistributionCenter
  (
     centerID VARCHAR(10) CHARACTER SET ASCII,
     name_    VARCHAR(50) CHARACTER SET utf8mb4 NOT NULL,
     phone    VARCHAR(12) CHARACTER SET ASCII,
     address  VARCHAR(100) CHARACTER SET utf8mb4,
     country  CHAR(2) CHARACTER SET ASCII DEFAULT 'US',
     website  VARCHAR(50) CHARACTER SET utf8mb4,
     PRIMARY KEY(centerID),
     CHECK (country REGEXP '^[A-Z]{2}$'),
```

```
    CHECK (phone REGEXP '^[0-9]{7,12}$')
  );

CREATE TABLE make
  (
    supplierID VARCHAR(10) CHARACTER SET ASCII,
    designerID VARCHAR(10) CHARACTER SET ASCII,
    setID      VARCHAR(10) CHARACTER SET ASCII,
    PRIMARY KEY(supplierID, designerID, setID),
    FOREIGN KEY(supplierID) REFERENCES Supplier(supplierID),
    FOREIGN KEY(designerID) REFERENCES Designer(designerID),
    FOREIGN KEY(setID) REFERENCES Set_(setID)
  );

CREATE TABLE contains_
  (
    setID       VARCHAR(10) CHARACTER SET ASCII,
    modelNumber VARCHAR(10) CHARACTER SET ASCII,
    count_      TINYINT UNSIGNED DEFAULT 1,
    PRIMARY KEY(setID, modelNumber),
    FOREIGN KEY(setID) REFERENCES Set_(setID),
    FOREIGN KEY(modelNumber) REFERENCES Model(modelNumber),
    CHECK (count_ > 0)
  );

CREATE TABLE describes
  (
    modelNumber VARCHAR(10) CHARACTER SET ASCII NOT NULL,
    sku         VARCHAR(10) CHARACTER SET ASCII,
    PRIMARY KEY(sku),
    FOREIGN KEY(modelNumber) REFERENCES Model(modelNumber),
    FOREIGN KEY(sku) REFERENCES Item(sku)
  );

CREATE TABLE canOrderFrom
  (
    centerID   VARCHAR(10) CHARACTER SET ASCII,
    supplierID VARCHAR(10) CHARACTER SET ASCII,
    leadTime   DOUBLE UNSIGNED,
    PRIMARY KEY(centerID, supplierID),
    FOREIGN KEY(centerID) REFERENCES DistributionCenter(centerID),
    FOREIGN KEY(supplierID) REFERENCES Supplier(supplierID)
  );

CREATE TABLE stocks
```

```
  (
     centerID VARCHAR(10) CHARACTER SET ASCII NOT NULL,
     sku      VARCHAR(10) CHARACTER SET ASCII,
     PRIMARY KEY(sku),
     FOREIGN KEY(centerID) REFERENCES DistributionCenter(centerID),
     FOREIGN KEY(sku) REFERENCES Item(sku)
  );

CREATE TABLE Chair
  (
     sku          VARCHAR(10) CHARACTER SET ASCII,
     numberOfLegs TINYINT UNSIGNED DEFAULT 4,
     hasCushion   BOOL DEFAULT false,
     hasArms      BOOL,
     backHeight   DOUBLE UNSIGNED,
     seatHeight   DOUBLE UNSIGNED,
     PRIMARY KEY(sku),
     FOREIGN KEY(sku) REFERENCES Item(sku),
     CHECK (numberOfLegs > 0),
     CHECK (backHeight > 0.0),
     CHECK (seatHeight > 0.0)
  );

CREATE TABLE Table_
  (
     sku           VARCHAR(10) CHARACTER SET ASCII,
     numberOfLegs  TINYINT UNSIGNED DEFAULT 4,
     numberOfSeats TINYINT UNSIGNED,
     shape         VARCHAR(30) CHARACTER SET utf8mb4,
     PRIMARY KEY(sku),
     FOREIGN KEY(sku) REFERENCES Item(sku),
     CHECK (numberOfLegs > 0),
     CHECK (numberOfSeats > 0)
  );

CREATE TABLE Desk
  (
     sku             VARCHAR(10) CHARACTER SET ASCII,
     angle           DOUBLE DEFAULT 0.0,
     numberOfDrawers TINYINT UNSIGNED,
     PRIMARY KEY(sku),
     FOREIGN KEY(sku) REFERENCES Item(sku),
     CHECK (angle > -360.0 AND angle < 360.0),
     CHECK (numberOfDrawers > 0)
  );
```

```
164
165 CREATE TABLE Stool
166   (
167      sku          VARCHAR(10) CHARACTER SET ASCII,
168      numberOfLegs TINYINT UNSIGNED,
169      hasCushion   BOOL,
170      hasSwivel    BOOL,
171      PRIMARY KEY(sku),
172      FOREIGN KEY(sku) REFERENCES Item(sku),
173      CHECK (numberOfLegs > 0)
174   );
175
176 CREATE TABLE Cabinet
177   (
178      sku                  VARCHAR(10) CHARACTER SET ASCII,
179      numberOfCompartments TINYINT UNSIGNED,
180      capacity             VARCHAR(30) CHARACTER SET utf8mb4,
181      PRIMARY KEY(sku),
182      FOREIGN KEY(sku) REFERENCES Item(sku),
183      CHECK (numberOfCompartments > 0)
184   );
185
186 CREATE TABLE Bedframe
187   (
188      sku    VARCHAR(10) CHARACTER SET ASCII,
189      size_  VARCHAR(30) CHARACTER SET utf8mb4,
190      depth_ DOUBLE,
191      PRIMARY KEY(sku),
192      FOREIGN KEY(sku) REFERENCES Item(sku)
193   );
194
195 CREATE TABLE features_Feature
196   (
197      modelNumber VARCHAR(10) CHARACTER SET ASCII,
198      description VARCHAR(50) CHARACTER SET utf8mb4,
199      count_      TINYINT UNSIGNED DEFAULT 1,
200      PRIMARY KEY(modelNumber, description),
201      FOREIGN KEY(modelNumber) REFERENCES Model(modelNumber),
202      CHECK (count_ > 0)
203   );
```

Data Counts

```
                    Supplier  50
                    Designer  50
                        Set_  100
                       Model  1000
                        Item  1000
          DistributionCenter  50
                        make  1000
                   contains_  991
                   describes  1000
                canOrderFrom  813
                      stocks  274
                       Chair  185
                      Table_  171
                        Desk  149
                       Stool  183
                     Cabinet  171
                    Bedframe  115
          features_Feature  8351
```

Sample Interactions

```
mysql> explain Bedframe;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| sku    | varchar(10) | NO   | PRI | NULL    |       |
| size_  | varchar(30) | YES  |     | NULL    |       |
| depth_ | double      | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> insert into Bedframe values (('6yNuvTHGL9', 'double twin', 5.9));
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql> insert into Bedframe values ('6yNuvTHGL9', 'double twin', 5.9);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint
 ↪  fails (`aaltman`.`bedframe`, CONSTRAINT `bedframe_ibfk_1` FOREIGN KEY (`sku`)
 ↪  REFERENCES `Item` (`sku`))
mysql> explain Item;
+-------------+-----------------+------+-----+---------+-------+
| Field       | Type            | Null | Key | Default | Extra |
+-------------+-----------------+------+-----+---------+-------+
| sku         | varchar(10)     | NO   | PRI | NULL    |       |
| length_     | double unsigned | YES  |     | NULL    |       |
| width       | double unsigned | YES  |     | NULL    |       |
```

```
| height      | double unsigned | YES  |     | NULL    |       |
| condition_  | varchar(30)     | YES  |     | NULL    |       |
| weightLimit | double unsigned | YES  |     | NULL    |       |
+-------------+-----------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> insert into Item values ('6yNuvTHGL9', 120.0, 70.0, 40.5, 'like new',
 ↪  276.89);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Bedframe values ('6yNuvTHGL9', 'double twin', 5.9);
Query OK, 1 row affected (0.00 sec)

mysql> select * from Item;
+------------+---------+-------+--------+------------+-------------+
| sku        | length_ | width | height | condition_ | weightLimit |
+------------+---------+-------+--------+------------+-------------+
| 6yNuvTHGL9 |     120 |    70 |   40.5 | like new   |      276.89 |
+------------+---------+-------+--------+------------+-------------+
1 row in set (0.00 sec)

mysql> select * from Bedframe;
+------------+-------------+--------+
| sku        | size_       | depth_ |
+------------+-------------+--------+
| 6yNuvTHGL9 | double twin |    5.9 |
+------------+-------------+--------+
1 row in set (0.00 sec)

mysql> delete from Bedframe select * from Bedframe;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
 ↪  corresponds to your MySQL server version for the right syntax to use near
 ↪  'select * from Bedframe' at line 1
mysql> delete from Bedframe;
Query OK, 1 row affected (0.00 sec)

mysql> delete from Item;
Query OK, 1 row affected (0.01 sec)
```

- We learned the proper syntax for `INSERT` and `DELETE` statements.
- We learned not to insert a value in a table that has a foreign key constraint before that constraint is satisfied.

Data Sources

We generated our data randomly, using the following C++ program (`random_gen/random_gen.cpp`), along with some initial data files drawn from `https://www.random.org`:

```cpp
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <fstream>
#include <iostream>
#include <string>
using namespace std;

int main() {
  srand((unsigned int)time(NULL));

  int i = 0;

  string sku[2000];

  ifstream skuList_file("skuList.txt");

  ifstream setIDs_file("setIDs.txt");

  ofstream set_data;
  ofstream output;
  ofstream contains_file;
  ofstream chairs_output;
  ofstream tables_output;
  ofstream desks_output;
  ofstream stools_output;
  ofstream cabinets_output;
  ofstream bedframes_output;
  ofstream make_file;
  ofstream canOrderFrom_file;
  string line;
  ifstream modelNumbers_file("modelNumbers.txt");
  string name1[18];
  string name2[18];
  string name3[6];
  string material[26];
  string upholstery[13];
  string color[25];
  string durability[10];
```

```
40    string modelNumbers[1000];
41
42    ifstream styles_file("styles.txt");
43
44    string setIDs[100];
45    string styles[100];
46
47    if (styles_file.is_open()) {
48      while (getline(styles_file, line)) {
49        styles[i] = line;
50        i++;
51        // cout << line << endl;
52      }
53      styles_file.close();
54    } else
55      cerr << "Unable to open styles file" << endl;
56
57    name1[0] = "Timeless";
58    name1[1] = "Futuristic";
59    name1[2] = "Contemporary";
60    name1[3] = "Homely";
61    name1[4] = "Ancient";
62    name1[5] = "Stylish";
63    name1[6] = "Eccentric";
64    name1[7] = "Large";
65    name1[8] = "Aged";
66    name1[9] = "Strong";
67    name1[10] = "Eclectic";
68    name1[11] = "Rustic";
69    name1[12] = "Modern";
70    name1[13] = "Revolutionary";
71    name1[14] = "Indian";
72    name1[15] = "German";
73    name1[16] = "Italian";
74    name1[17] = "Norwegian";
75
76    name2[0] = "Angular";
77    name2[1] = "Rounded";
78    name2[2] = "Jagged";
79    name2[3] = "Hefty";
80    name2[4] = "Light";
81    name2[5] = "Chunky";
82    name2[6] = "Macho";
83    name2[7] = "Thin";
84    name2[8] = "Badass";
```

```
    name2[9] = "Ladylike";
    name2[10] = "Slender";
    name2[11] = "Wimpy";
    name2[12] = "Oblique";
    name2[13] = "Bowed";
    name2[14] = "Gaunt";
    name2[15] = "Meager";
    name2[16] = "Fat";
    name2[17] = "Genteel";

    name3[0] = "Chair";
    name3[1] = "Table";
    name3[2] = "Desk";
    name3[3] = "Stool";
    name3[4] = "Cabinet";
    name3[5] = "Bedframe";

    material[0] = "Ash";
    material[1] = "Cherry";
    material[2] = "Maple";
    material[3] = "Birch";
    material[4] = "Teak";
    material[5] = "Hickory";
    material[6] = "Oak";
    material[7] = "Walnut";
    material[8] = "Aluminum";
    material[9] = "Steel";
    material[10] = "Beech";
    material[11] = "Alder";
    material[12] = "Elm";
    material[13] = "Pine";
    material[14] = "Cottonwood";
    material[15] = "Hemlock";
    material[16] = "Fir";
    material[17] = "Cedar";
    material[18] = "Balsa";
    material[19] = "Magnesium Alloy";
    material[20] = "Coast Redwood";
    material[21] = "Afzelia";
    material[22] = "Ebony";
    material[23] = "Lindens";
    material[24] = "Purpleheart";
    material[25] = "Aspen";

    upholstery[0] = "Linen";
```

```
    upholstery[1] = "Leather";
    upholstery[2] = "Cotton";
    upholstery[3] = "Wool";
    upholstery[4] = "Cotton Blend";
    upholstery[5] = "Vinyl";
    upholstery[6] = "Silk";
    upholstery[7] = "Acetate";
    upholstery[8] = "Acrylic";
    upholstery[9] = "Nylon";
    upholstery[10] = "Olefin";
    upholstery[11] = "Polyester";
    upholstery[12] = "Rayon";

    color[0] = "Blue";
    color[1] = "Green";
    color[2] = "Red";
    color[3] = "White";
    color[4] = "Black";
    color[5] = "Yellow";
    color[6] = "Grey";
    color[7] = "Orange";
    color[8] = "Purple";
    color[9] = "Pink";
    color[10] = "Violet";
    color[11] = "Magenta";
    color[12] = "Gold";
    color[13] = "Cyan";
    color[14] = "Turquoise";
    color[15] = "Lavender";
    color[16] = "Maroon";
    color[17] = "Olive";
    color[18] = "Indigo";
    color[19] = "Tan";
    color[20] = "Salmon";
    color[21] = "Sky Blue";
    color[22] = "Teal";
    color[23] = "Coral";
    color[24] = "Silver";

    durability[0] = "Very Strong";
    durability[1] = "Strong";
    durability[2] = "Somewhat Strong";
    durability[3] = "Very Sturdy";
    durability[4] = "Sturdy";
    durability[5] = "Somewhat Sturdy";
```

```
    durability[6] = "Wobbly";
    durability[7] = "Somewhat Wobbly";
    durability[8] = "Very Wobbly";
    durability[9] = "Indestructable";

    output.open("model_data.csv");

    chairs_output.open("chairs.csv");
    tables_output.open("tables.csv");
    desks_output.open("desks.csv");
    stools_output.open("stools.csv");
    cabinets_output.open("cabinets.csv");
    bedframes_output.open("bedframes.csv");

    string item;
    string preName;
    string postName;
    string current_material;
    string current_upholstery;
    string current_durability;
    string current_color;

    i = 0;

    if (skuList_file.is_open()) {
      while (getline(skuList_file, line)) {
        sku[i] = line;
        i++;
      }
      skuList_file.close();
    } else
      cerr << "Unable to open skuList file" << endl;

    i = 0;
    // Write the modelNumbers into a file
    if (modelNumbers_file.is_open()) {
      while (getline(modelNumbers_file, line)) {
        modelNumbers[i] = line;

        int current_item = rand() % 6;

        preName = name1[rand() % 18];
        postName = name2[rand() % 18];
        current_material = material[rand() % 25];
        current_upholstery = upholstery[rand() % 12];
```

```
        current_durability = durability[rand() % 10];
        current_color = color[rand() % 25];
        item = name3[current_item];

        switch (current_item + 1) {
        case 1:
          chairs_output << line << ",";
          chairs_output << sku[i] << ",";
          i++;
          chairs_output << rand() % 3 + 3 << ",";
          chairs_output << rand() % 2 << ",";
          chairs_output << rand() % 2 << ",";
          chairs_output << rand() % 12 + 24 << ",";
          chairs_output << rand() % 24 + 24 << endl;
          break;
        case 2:
          tables_output << line << ",";
          tables_output << sku[i] << ",";
          i++;
          tables_output << rand() % 6 + 4 << ",";
          tables_output << rand() % 8 + 4 << ",";
          if (rand() % 2 == 1) {
            tables_output << "Round" << endl;
          } else {
            tables_output << "Rectangular" << endl;
          }
          break;
        case 3:
          desks_output << line << ",";
          desks_output << sku[i];
          i++;
          desks_output << rand() % 45 + 0 << ",";
          desks_output << rand() % 6 + 1 << endl;
          break;
        case 4:
          stools_output << line << ",";
          stools_output << sku[i] << ",";
          i++;
          stools_output << rand() % 3 + 3 << ",";
          stools_output << rand() % 2 << ",";
          stools_output << rand() % 2 << endl;
          break;
        case 5:
          cabinets_output << line << ",";
          cabinets_output << sku[i] << ",";
```

```
265           i++;
266           cabinets_output << rand() % 6 + 1 << ",";
267           cabinets_output << rand() % 10 + 1 << endl;
268           break;
269         case 6:
270           bedframes_output << line << ",";
271           bedframes_output << sku[i] << ",";
272           switch (rand() % 6 + 1) {
273           case 1:
274             bedframes_output << "Twin"
275                              << ",";
276             break;
277           case 2:
278             bedframes_output << "Twin XL"
279                              << ",";
280             break;
281           case 3:
282             bedframes_output << "Full"
283                              << ",";
284             break;
285           case 4:
286             bedframes_output << "Queen"
287                              << ",";
288             break;
289           case 5:
290             bedframes_output << "King"
291                              << ",";
292             break;
293           case 6:
294             bedframes_output << "California King"
295                              << ",";
296             break;
297           default:
298             cerr << "bad case in switch statement!!!" << endl;
299           }
300           bedframes_output << rand() % 24 + 24 << endl;
301           break;
302         default:
303           cerr << "bad case in switch statement!!!" << endl;
304         }
305
306         output << line << ",";
307         output << preName << " " << postName << " " << item << ",";
308         output << current_material << ",";
309         output << current_upholstery << ",";
```

```
      output << current_durability << ",";
      output << current_color << endl;
    }
    modelNumbers_file.close();
  } else
    cerr << "Unable to open modelNumbers file";

  ofstream output_file;

  ifstream designerIDs_file("designerIDs.txt");
  ifstream firstNames_file("firstNames.txt");
  ifstream lastNames_file("lastNames.txt");
  ifstream phones_file("phones.txt");
  ifstream designerAddresses_file("designerAddresses.txt");
  ifstream countries_file("countries.txt");
  ifstream designFocuses_file("designFocuses.txt");

  string designerIDs[50];
  string firstNames[50];
  string lastNames[50];
  string phones[50];
  string designerAddresses[50];
  string countries[249];
  string websites[50];
  string designFocuses[11];
  string domainSuffix[5];

  domainSuffix[0] = ".com";
  domainSuffix[1] = ".net";
  domainSuffix[2] = ".org";
  domainSuffix[3] = ".biz";
  domainSuffix[4] = ".info";

  i = 0;

  // Read "designerIDs.txt" into string array

  if (designerIDs_file.is_open()) {
    while (getline(designerIDs_file, line)) {
      designerIDs[i] = line;
      i++;
    }
    designerIDs_file.close();
  } else
    cerr << "Unable to open ID file" << endl;
```

```
355
356   i = 0;
357
358   // Read "firstNames.txt" into string array
359
360   if (firstNames_file.is_open()) {
361     while (getline(firstNames_file, line)) {
362       firstNames[i] = line;
363       i++;
364     }
365     firstNames_file.close();
366   } else
367     cerr << "Unable to open firstNames file" << endl;
368
369   i = 0;
370
371   // Read "lastNames.txt" into string array
372
373   if (lastNames_file.is_open()) {
374     while (getline(lastNames_file, line)) {
375       lastNames[i] = line;
376       i++;
377     }
378     lastNames_file.close();
379   } else
380     cerr << "Unable to open lastNames file" << endl;
381
382   i = 0;
383
384   // Read "phones.txt" into string array
385
386   if (phones_file.is_open()) {
387     while (getline(phones_file, line)) {
388       phones[i] = line;
389       i++;
390     }
391     phones_file.close();
392   } else
393     cerr << "Unable to open phones file" << endl;
394
395   i = 0;
396
397   // Read "designerAddresses.txt into string array
398
399   if (designerAddresses_file.is_open()) {
```

```
    while (getline(designerAddresses_file, line)) {
      designerAddresses[i] = line;
      i++;
    }
    designerAddresses_file.close();
  } else
    cerr << "Unable to open Addresses file" << endl;

  i = 0;

  // Read "countries.txt" into string array
  if (countries_file.is_open()) {
    while (getline(countries_file, line)) {
      countries[i] = line;
      i++;
    }
    countries_file.close();
  } else
    cerr << "Unable to open countries file" << endl;

  i = 0;

  // Create website addresses based on names, save them into websites array
  while (i < 50) {
    websites[i] =
        "www." + firstNames[i] + lastNames[i] + domainSuffix[rand() % 5];
    i++;
  }

  i = 0;

  // Read "designFocuses.txt" into string array

  if (designFocuses_file.is_open()) {
    while (getline(designFocuses_file, line)) {
      designFocuses[i] = line;
      i++;
    }
    designFocuses_file.close();
  } else
    cerr << "Unable to open designFocuses file" << endl;

  i = 0;
  // Write to output file
  // use a loop to write output to a file
```

```
445   // SUBTASKS
446   // write designerID on a line, end line
447   // write a first name then a last name on a line, end line
448   // write a phone number on a line, end line
449   // write an address on a line, end line
450   // write a random country on a line, end line
451   // write a website on a line, end line
452   // write a random design focus on a line, end line
453   // end line for spacing
454   output_file.open("designer_data.csv");
455
456   while (i < 50) {
457     output_file << designerIDs[i] << ",";
458     output_file << firstNames[i] << " " << lastNames[i] << ",";
459     output_file << phones[i] << ",";
460     output_file << "\"" << designerAddresses[i] << "\""
461                 << ",";
462     output_file << countries[rand() % 249] << ",";
463     output_file << websites[i] << ",";
464     output_file << designFocuses[rand() % 11] << endl;
465     i++;
466   }
467   output_file.close();
468
469   i = 0;
470
471   string supplierIDs[50];
472   string supplierNames[50];
473   string supplierWebsiteNames[50];
474   string addresses[50];
475
476   // supplierIDs.txt
477   ifstream supplierIDs_file("supplierIDs.txt");
478   // names.txt
479   ifstream supplierNames_file("supplierNames.txt");
480   // supplierWebsiteNames.txt
481   ifstream supplierWebsiteNames_file("supplierWebsiteNames.txt");
482   // addresses.txt
483   ifstream addresses_file("addresses.txt");
484
485   // Read "supplierIDs.txt" into an array
486
487   if (supplierIDs_file.is_open()) {
488     while (getline(supplierIDs_file, line)) {
489       supplierIDs[i] = line;
```

```
      i++;
    }
    supplierIDs_file.close();
  } else
    cerr << "Unable to open supplierIDs file" << endl;

  i = 0;

  // Read "supplierNames.txt" to names array

  if (supplierNames_file.is_open()) {
    while (getline(supplierNames_file, line)) {
      supplierNames[i] = line;
      i++;
    }
    supplierNames_file.close();
  } else
    cerr << "Unable to open names file" << endl;

  i = 0;

  // read "supplierWebsiteNames.txt" into an array

  if (supplierWebsiteNames_file.is_open()) {
    while (getline(supplierWebsiteNames_file, line)) {
      supplierWebsiteNames[i] = line;
      i++;
    }
    supplierWebsiteNames_file.close();
  } else
    cerr << "Unable to open supplierWebsiteNames file" << endl;

  i = 0;

  // Read "addresses.txt" to addresses array

  if (addresses_file.is_open()) {
    while (getline(addresses_file, line)) {
      addresses[i] = line;
      i++;
    }
    addresses_file.close();
  } else
    cerr << "Unable to open addresses file" << endl;

```

```
535   i = 0;
536
537   // output data to supplier_data.txt
538   // supplierIDs
539   // name
540   // phone
541   // address
542   // country
543   // website
544   output_file.open("supplier_data.csv");
545   while (i < 50) {
546     output_file << supplierIDs[i] << ",";
547     output_file << supplierNames[i] << ",";
548     output_file << phones[i] << ",";
549     output_file << "\"" << addresses[i] << "\""
550                 << ",";
551     output_file << countries[rand() % 249] << ",";
552     output_file << "www." << supplierWebsiteNames[i]
553                 << domainSuffix[rand() % 5];
554     output_file << endl;
555     i++;
556   }
557   output_file.close();
558
559   i = 0;
560
561   string centerIDs[50];
562   string centerNames[50];
563   string centerWebsiteNames[50];
564
565   // centerIDs.txt
566   ifstream centerIDs_file("centerIDs.txt");
567   // names.txt
568   ifstream centerNames_file("centerNames.txt");
569   // centerWebsiteNames.txt
570   ifstream centerWebsiteNames_file("centerWebsiteNames.txt");
571
572   // Read "centerIDs.txt" into an array
573
574   if (centerIDs_file.is_open()) {
575     while (getline(centerIDs_file, line)) {
576       centerIDs[i] = line;
577       i++;
578     }
579     centerIDs_file.close();
```

```
580   } else
581     cerr << "Unable to open centerIDs file" << endl;
582
583   i = 0;
584
585   // Read "centerNames.txt" to names array
586
587   if (centerNames_file.is_open()) {
588     while (getline(centerNames_file, line)) {
589       centerNames[i] = line;
590       i++;
591     }
592     centerNames_file.close();
593   } else
594     cerr << "Unable to open names file" << endl;
595
596   i = 0;
597
598   // read "centerWebsiteNames.txt" into an array
599
600   if (centerWebsiteNames_file.is_open()) {
601     while (getline(centerWebsiteNames_file, line)) {
602       centerWebsiteNames[i] = line;
603       i++;
604     }
605     centerWebsiteNames_file.close();
606   } else
607     cerr << "Unable to open centerWebsiteNames file" << endl;
608
609   i = 0;
610
611   // output data to center_data.txt
612   // centerIDs
613   // name
614   // phone
615   // address
616   // country
617   // website
618   output_file.open("center_data.csv");
619   while (i < 50) {
620     output_file << centerIDs[i] << ",";
621     output_file << centerNames[i] << ",";
622     output_file << phones[i] << ",";
623     output_file << "\"" << addresses[i] << "\""
624                 << ",";
```

```
625       output_file << countries[rand() % 249] << ",";
626       output_file << "www." << centerWebsiteNames[i] << domainSuffix[rand() % 5];
627       output_file << endl;
628       i++;
629     }
630     output_file.close();
631
632     ifstream modelNumber_file("modelNumbers.txt");
633
634     ofstream modelNumberToSku_file;
635     ofstream item_file;
636
637     i = 0;
638
639     // Take in "modelNumbers.txt" and put into array
640
641     if (modelNumber_file.is_open()) {
642       while (getline(modelNumber_file, line)) {
643         modelNumbers[i] = line;
644         i++;
645       }
646       modelNumber_file.close();
647     } else
648       cerr << "Unable to open modelNumber file" << endl;
649
650     i = 0;
651     // output a file that assigns each modelNumber an sku
652
653     modelNumberToSku_file.open("modelNumberToSku.csv");
654
655     while (i < 1000) {
656       modelNumberToSku_file << modelNumbers[i] << "," << sku[i] << endl;
657       i++;
658     }
659     i = 0;
660
661     // output "item_data.csv"
662
663     item_file.open("item_data.csv");
664
665     while (i < 1000) {
666       item_file << sku[i] << "," << rand() % 24 + 48 << "," << rand() % 24 + 48
667                 << "," << rand() % 24 + 48 << ","
668                 << "New"
669                 << "," << rand() % 200 + 300 << endl;
```

```
    i++;
  }
  i = 0;

  if (setIDs_file.is_open()) {
    while (getline(setIDs_file, line)) {
      setIDs[i] = line;
      i++;
    }
    setIDs_file.close();
  } else
    cerr << "Unable to open setIDs file" << endl;

  i = 0;

  set_data.open("set_data.csv");

  while (i < 100) {
    set_data << setIDs[i] << ",";
    set_data << styles[i] << " set"
             << ",";
    set_data << rand() % 33 + 1985 << ",";
    set_data << i << ",";
    set_data << styles[i] << endl;
    i++;
  }

  contains_file.open("contains_data.csv");
  i = 0;
  while (i < 1000) {
    contains_file << setIDs[rand() % 100] << ",";
    contains_file << modelNumbers[rand() % 1000] << ",";
    contains_file << (rand() % 6 + 1) << endl;
    i++;
  }
  contains_file.close();

  i = 0;

  make_file.open("make_data.csv");

  while (i < 1000) {
    make_file << supplierIDs[rand() % 50] << ",";
    make_file << designerIDs[rand() % 50] << ",";
    make_file << setIDs[rand() % 100] << endl;
```

```
715       i++;
716     }
717
718     make_file.close();
719
720     i = 0;
721
722     canOrderFrom_file.open("canOrderFrom_data.csv");
723
724     while (i < 1000) {
725       canOrderFrom_file << centerIDs[rand() % 50] << ",";
726       canOrderFrom_file << supplierIDs[rand() % 50] << ",";
727       canOrderFrom_file << (rand() % 30 + 1) << endl;
728
729       i++;
730     }
731     canOrderFrom_file.close();
732
733     ofstream stocks_file;
734
735     stocks_file.open("stocks_data.csv");
736
737     for (int l = 0; l < 50; l++) {
738       int count = rand() % 15 + 1;
739       for (int x = 1; x < count; x++) {
740         stocks_file << centerIDs[l] << "," << sku[rand() % 1000] << endl;
741       }
742     }
743     stocks_file.close();
744
745     string features[10];
746
747     features[0] = "Fancy Knobs";
748     features[1] = "Carved Inlays";
749     features[2] = "Ivory Handles";
750     features[3] = "Claw Feet";
751     features[4] = "Gold Inlaid Designs";
752     features[5] = "Fancy Molding";
753     features[6] = "Gold Hinges";
754     features[7] = "Padded Feet";
755     features[8] = "Studded Corners";
756     features[9] = "Textured";
757     ofstream features_file;
758
759     features_file.open("features_data.csv");
```

```
760
761   for (int l = 0; l < 1000; l++) {
762     for (int x = 0; x < 10; x++) {
763       int count = rand() % 6;
764       if (count > 0) {
765         features_file << modelNumbers[l] << "," << features[x] << "," << count
766                       << endl;
767       }
768     }
769   }
770
771   features_file.close();
772
773   return 0;
774 }
```

We used the command syntax `load data local infile {CSV file name} ignore into table {table name} fields terminated by ',' optionally enclosed by '"';` to load the data into each table from the corresponding CSV file, which took care of duplicate entries by ignoring them; all other constraints were satisfied by construction due to the way the generation program was written and run.

Data Samples

```
1 mysql> select * from Supplier limit 10;
2 +------------+-------------------------------------+------------+------------------
   ↪ ----------------------------+---------+--------------------------------------
   ↪ --+
3 | supplierID | name_                               | phone      | address
   ↪                              | country | website
   ↪    |
4 +------------+-------------------------------------+------------+------------------
   ↪ ----------------------------+---------+--------------------------------------
   ↪ --+
5 | 06DeQdaf4X | The Snowy Pencil Leather Company    | 8039090134 | 379 Virginia St.
   ↪  Niagara Falls, NY 14304      | TR      | www.TheSnowyPencilLeatherCompany.biz
   ↪    |
6 | 0OaJOGFGdD | The Cloudy Chicken Fabrics Company | 8749357327 | 8247 S. Hamilton
   ↪  Drive Cedar Rapids, IA 52402 | TO      |
   ↪  www.TheCloudyChickenFabricsCompany.info |
7 | 0x1V8UVLqh | The Deep Lamp Lumber Company        | 7463153612 | 72 Somerset Lane
   ↪  Ypsilanti, MI 48197          | AQ      | www.TheDeepLampLumberCompany.info
   ↪    |
```

```
| 3H7BBv3j7C | Tall Mouse Supplier                | 4123072387 | 9444 Blackburn Lane Wenatchee, WA 98801       | PA      | www.TallMouseSupplier.org      |
| 3LnmekGoPa | The Happy Chinchilla Metal Company | 9115174479 | 50 Catherine Lane Huntsville, AL 35803        | ZM      | www.TheHappyChinchillaMetalCompany.com   |
| 45AVHG6SDL | Big Light-Switch Metal             | 2219847475 | 9916 Bridgeton Ave. Austin, MN 55912          | AQ      | www.BigLight-SwitchMetal.org      |
| 4hGbj2aBVR | Red Baboon Metal                   | 8433984003 | 860 Glen Ridge Rd. Whitestone, NY 11357       | BT      | www.RedBaboonMetal.net      |
| 4xp0ETTkc8 | The White Skunk Metal Company      | 2695261566 | 52 Deerfield Lane Woodhaven, NY 11421         | DJ      | www.TheWhiteSkunkMetalCompany.info   |
| 5Hq7fF9aK0 | The Opaque Pear Supplier Company   | 9201886129 | 7254 Hickory Ave. Centreville, VA 20120       | CI      | www.TheOpaquePearSupplierCompany.org   |
| 5SKHZGJsjV | Transparent Cherry Lumber          | 4623416776 | 1 Redwood Rd. Roanoke, VA 24012               | ZW      | www.TransparentCherryLumber.org      |
+------------+------------------------------------+------------+------------------------------+---------+---------------------------------------+
10 rows in set (0.00 sec)

mysql> select * from Designer limit 10;
+------------+-----------------+------------+-----------------------------------+--------------+---------+-------------------------+--------------+
| designerID | name_           | phone      | address                           | country | website                 | designFocus  |
+------------+-----------------+------------+-----------------------------------+--------------+---------+-------------------------+--------------+
| 3lejckzNYS | Lyndia Butler   | 7034816719 | 241 Morris Dr. Bowling Green, KY 42101            | PE      | www.LyndiaButler.org    | Mid-Century  |
| 4MVbu2iI15 | Mellissa Rich   | 4623416776 | 4 Garden Rd. Dunedin, FL 34698                | AW      | www.MellissaRich.biz    | Scandinavian |
| 7RYpyw9es0 | Casimira Carney | 1437523438 | 8498 Young Street Oklahoma City, OK 73112         | ML      | www.CasimiraCarney.com  | Industrial   |
| 7ZZbCsXnv0 | Charmaine Pineda | 4204843909 | 497 Young Lane Panama City, FL 32404              | NL      | www.CharmainePineda.org | Rococo       |
| A7oA1v9Ax1 | Renee Holmes    | 2589423308 | 78 Santa Clara Drive Huntsville, AL 35803         | ZW      | www.ReneeHolmes.org     | Modern       |
| AZadqlHsUN | Elia Melton     | 7421854946 | 519 Bayberry Ave. Bayside, NY 11361               | SX      | www.EliaMelton.org      | Industrial   |
```

```
| b9yOGUx3pl | Jeni Wilson      | 5416771400 | 215 Birchwood Ave. Boston, MA 02127
 ↪                | KR      | www.JeniWilson.net      | Industrial   |
| bBMMbiXWX2 | Ludivina Hunt    | 7071474222 | 9120 Santa Clara St. Huntington
 ↪  Station, NY 11746 | GP      | www.LudivinaHunt.net    | Rococo       |
| bDcoTRYgku | Sierra Novak     | 2695261566 | 155 N. Elm Street Rego Park, NY
 ↪  11374             | VN      | www.SierraNovak.org     | Scandinavian |
| BejwYSNzm7 | Claribel Vasquez | 6712497760 | 256 Water Ave. Shelton, CT 06484
 ↪                | LA      | www.ClaribelVasquez.org | Eclecticism  |
+------------+------------------+------------+-----------------------------------
 ↪  --------------+---------+-------------------------+--------------+
10 rows in set (0.00 sec)

mysql> select * from Set_ limit 10;
+------------+-----------------+-------------+----------------------+-------------+
| setID      | name_           | catalogYear | catalogNumber        | style_      |
+------------+-----------------+-------------+----------------------+-------------+
| 0naFKFb6se | dizzy set       |        2013 | 00000000000000000029 | dizzy       |
| 0nZQJHrEtZ | fowl set        |        1992 | 00000000000000000013 | fowl        |
| 0To5uOwhgK | spectacular set |        1986 | 00000000000000000040 | spectacular |
| 1JtLYWK555 | receive set     |        1985 | 00000000000000000070 | receive     |
| 5JnQxcSlvJ | earthquake set  |        1985 | 00000000000000000066 | earthquake  |
| 6eHdxhCYK7 | fish set        |        2001 | 00000000000000000000 | fish        |
| 73l1iRUEvw | capable set     |        2014 | 00000000000000000071 | capable     |
| 73X8UvpzrJ | slip set        |        1998 | 00000000000000000093 | slip        |
| 8blchiMxYL | right set       |        2000 | 00000000000000000038 | right       |
| 980pBKGuMU | file set        |        2003 | 00000000000000000083 | file        |
+------------+-----------------+-------------+----------------------+-------------+
10 rows in set (0.00 sec)

mysql> select * from Model limit 10;
+-------------+--------------------------+----------+------------+---------------
 ↪  -+---------+
| modelNumber | name_                    | material | upholstery | durability
 ↪   | color   |
+-------------+--------------------------+----------+------------+---------------
 ↪  -+---------+
| 00aWzohu8g  | Ancient Genteel Desk     | Ebony    | Wool       | Very Strong
 ↪   | Orange  |
| 02SVZ28q47  | Ancient Angular Stool    | Hemlock  | Nylon      | Very Wobbly
 ↪   | Silver  |
| 0aBIk36Hs2  | Contemporary Hefty Chair | Afzelia  | Acetate    | Very Strong
 ↪   | Indigo  |
| 0JCfsD5Pck  | Futuristic Rounded Stool | Maple    | Linen      | Sturdy
 ↪   | Grey    |
```

```
| 0KSfx7M1IN  | Contemporary Fat Desk     | Elm      | Cotton     | Very Wobbly
↪   | Teal    |
| 0KSjIO5LDu  | Eclectic Bowed Desk       | Cherry   | Cotton     | Sturdy
↪   | Black   |
| 0mp9PGPdKw  | Futuristic Bowed Bedframe | Balsa    | Linen      | Indestructable
↪   | Gold    |
| 0T2PaDHCEw  | Contemporary Light Chair  | Pine     | Acetate    | Somewhat
↪  Wobbly | Gold    |
| 0uzi2lU3dA  | Large Wimpy Table         | Fir      | Polyester  | Very Sturdy
↪   | Olive   |
| 0vhyAaLCYg  | Italian Rounded Bedframe  | Pine     | Cotton     | Indestructable
↪   | Magenta |
+-------------+---------------------------+----------+------------+----------------┘
↪  -+---------+
10 rows in set (0.00 sec)

mysql> select * from Item limit 10;
+------------+---------+-------+--------+------------+-------------+
| sku        | length_ | width | height | condition_ | weightLimit |
+------------+---------+-------+--------+------------+-------------+
| 09S4mRvuGE |      61 |    66 |     51 | New        |         309 |
| 0EbNx7hvlO |      64 |    58 |     48 | New        |         415 |
| 0EKsH99oc8 |      64 |    67 |     58 | New        |         342 |
| 0eLPBgmxXc |      67 |    57 |     53 | New        |         340 |
| 0F6T8XT2R1 |      50 |    59 |     53 | New        |         310 |
| 0hMisTAJ6K |      66 |    60 |     68 | New        |         311 |
| 0RzRFaHBm4 |      48 |    50 |     69 | New        |         350 |
| 0snow42N9H |      62 |    52 |     56 | New        |         325 |
| 0t5AqpGvtk |      63 |    65 |     65 | New        |         372 |
| 0WxtLk4cbX |      57 |    49 |     54 | New        |         302 |
+------------+---------+-------+--------+------------+-------------+
10 rows in set (0.00 sec)

mysql> select * from DistributionCenter limit 10;
+------------+-----------------------------------------------+------------+---------┘
↪  ----------------------------------+---------+------------------------------┘
↪  -------------------+
| centerID   | name_                                         | phone      | address
↪                                      | country | website
↪                     |
+------------+-----------------------------------------------+------------+---------┘
↪  ----------------------------------+---------+------------------------------┘
↪  -------------------+
```

```
| 0gkU23fhbT | Sub-Zero Duck Warehouse                        | 2738728093 | 7083 Green Court Pembroke Pines, FL 33028   | LA      |
| www.Sub-ZeroDuckWarehouse.com                      |
| 0mslNHYadG | The Complicated Hamster Distribution Company | 8256624098 | 904 Creekside St. Pottstown, PA 19464       | UG      |
| www.TheComplicatedHamsterDistributionCompany.net |
| 3D5zIxPUZv | Opaque Frog Storage                            | 1046485315 | 92 W. Princeton Rd. Long Beach, NY 11561    | KM      | www.OpaqueFrogStorage.info |
| 4XdNKanfY1 | The White Pear Warehouse Company               | 8485622783 | 9810 Edgefield St. Natick, MA 01760         | WS      |
| www.TheWhitePearWarehouseCompany.com               |
| 5RDZWNp0Ym | The Acute Turtle Warehouse Company             | 7003923806 | 99 Lower River Drive Mount Vernon, NY 10550 | KE      |
| www.TheAcuteTurtleWarehouseCompany.com             |
| 8ECOfFZSgC | The Grey Squirrel Distribution Company         | 1437523438 | 827 Dogwood Ave. Saginaw, MI 48601          | LC      |
| www.TheGreySquirrelDistributionCompany.biz         |
| a1Xzp6g1lw | Cold Fish Distributing                         | 8511426114 | 69 Bay Meadows Lane Bridgeton, NJ 08302     | KG      | www.ColdFishDistributing.com |
| aD4TjXPcH4 | Joey Distribution                              | 5911329769 | 937 Cypress Street Butte, MT 59701          | NU      | www.JoeyDistribution.org |
| At7FkrAz5m | Piping Bull Distributing                       | 9201886129 | 7254 Hickory Ave. Centreville, VA 20120     | EG      |
| www.PipingBullDistributing.net                     |
| beYPqnGi9f | Sub-Zero Lemon Warehouse                       | 7463153612 | 72 Somerset Lane Ypsilanti, MI 48197        | FJ      |
| www.Sub-ZeroLemonWarehouse.com                     |
+------------+------------------------------------------------+------------+---------+-------------------------------------+---------+-----------------------------+-------------------+
10 rows in set (0.00 sec)

mysql> select * from make limit 10;
+------------+------------+------------+
| supplierID | designerID | setID      |
+------------+------------+------------+
| 3H7BBv3j7C | 3lejckzNYS | jYIzpoPJ0v |
| 4hGbj2aBVR | 3lejckzNYS | pzZzDyittU |
| 5SKHZGJsjV | 3lejckzNYS | ajPkfYW6Rd |
| Dlq94L7rPh | 3lejckzNYS | LPi87bcU5k |
| dWuUGFOe1b | 3lejckzNYS | Auae4YJ0eg |
| ldbBahbqLj | 3lejckzNYS | MlgnC1JEJn |
```

```
| s9Fptw806W | 3lejckzNYS | XrPPnXrnJ3 |
| umALkT56Xy | 3lejckzNYS | grkjTJPNEP |
| 4hGbj2aBVR | 4MVbu2iI15 | x0XpZD2q1n |
| 4xp0ETTkc8 | 4MVbu2iI15 | aI8k5G0Fqi |
+------------+------------+------------+
10 rows in set (0.00 sec)

mysql> select * from contains_ limit 10;
+------------+-------------+--------+
| setID      | modelNumber | count_ |
+------------+-------------+--------+
| 0naFKFb6se | hv5bmT6oG9  |      4 |
| 0naFKFb6se | JnxJCLw8rc  |      2 |
| 0naFKFb6se | msWH7nqeNv  |      3 |
| 0naFKFb6se | RfBTVUqoof  |      3 |
| 0naFKFb6se | UhCgTjSXvT  |      2 |
| 0nZQJHrEtZ | H7AK7JFI2z  |      5 |
| 0nZQJHrEtZ | kphIGiILcS  |      6 |
| 0nZQJHrEtZ | lqKAZ0fWr5  |      3 |
| 0nZQJHrEtZ | OLFH7SPI6S  |      3 |
| 0nZQJHrEtZ | ouZNXv6e6y  |      5 |
+------------+-------------+--------+
10 rows in set (0.00 sec)

mysql> select * from describes limit 10;
+-------------+------------+
| modelNumber | sku        |
+-------------+------------+
| 00aWzohu8g  | g3v6PqXumq |
| 02SVZ28q47  | KRp6uvT5Nf |
| 0aBIk36Hs2  | TBraGAJvxH |
| 0JCfsD5Pck  | 4FsRw9nHzb |
| 0KSfx7M1IN  | GeGjTFxfw7 |
| 0KSjIO5LDu  | Xrrwxdca1D |
| 0mp9PGPdKw  | fa5G2jkkGG |
| 0T2PaDHCEw  | HIz1MJXQfO |
| 0uzi2lU3dA  | UDYoo8F309 |
| 0vhyAaLCYg  | PDhd8Kn8rC |
+-------------+------------+
10 rows in set (0.00 sec)

mysql> select * from canOrderFrom limit 10;
+------------+------------+----------+
| centerID   | supplierID | leadTime |
+------------+------------+----------+
```

```
| 0gkU23fhbT | 06DeQdaf4X |       28 |
| 0gkU23fhbT | 0OaJOGFGdD |        5 |
| 0gkU23fhbT | 3LnmekGoPa |       28 |
| 0gkU23fhbT | 45AVHG6SDL |       17 |
| 0gkU23fhbT | 5SKHZGJsjV |        1 |
| 0gkU23fhbT | 7Hh3kV9mIX |       29 |
| 0gkU23fhbT | 7n94AZBOkB |       28 |
| 0gkU23fhbT | 9kQZDUILDP |       28 |
| 0gkU23fhbT | dWuUGFOe1b |       12 |
| 0gkU23fhbT | EMLwuAKAEG |       26 |
+------------+------------+----------+
10 rows in set (0.00 sec)

mysql> select * from stocks limit 10;
+------------+------------+
| centerID   | sku        |
+------------+------------+
| 0gkU23fhbT | 6wdiSd4mc0 |
| 0gkU23fhbT | fB8CTGobFM |
| 0gkU23fhbT | fdUqKPeyDL |
| 0gkU23fhbT | INtjFxGvYo |
| 0gkU23fhbT | mISzbMdNjf |
| 0gkU23fhbT | nsfiUC8eEg |
| 0gkU23fhbT | OMeAaXgdtd |
| 0gkU23fhbT | RzHfSIqXgP |
| 0mslNHYadG | C96OgY6RFU |
| 0mslNHYadG | fSiM5um0Ex |
+------------+------------+
10 rows in set (0.00 sec)

mysql> select * from Chair limit 10;
+------------+--------------+------------+---------+------------+------------+
| sku        | numberOfLegs | hasCushion | hasArms | backHeight | seatHeight |
+------------+--------------+------------+---------+------------+------------+
| 0RzRFaHBm4 |            4 |          0 |       0 |         30 |         37 |
| 171xmqOt46 |            4 |          1 |       1 |         27 |         25 |
| 1mtTrTYoiN |            4 |          0 |       1 |         25 |         42 |
| 1OHH0exARc |            5 |          0 |       1 |         32 |         43 |
| 1xIdirGiRI |            5 |          0 |       1 |         25 |         43 |
| 2DUG1Jtxtm |            5 |          0 |       0 |         26 |         32 |
| 2oXo7jOoTO |            4 |          0 |       1 |         29 |         43 |
| 2PijUpIwmL |            4 |          0 |       0 |         27 |         26 |
| 30B5Vsjt9F |            4 |          1 |       0 |         25 |         41 |
| 3c4v9htJja |            4 |          1 |       0 |         25 |         25 |
+------------+--------------+------------+---------+------------+------------+
```

```
10 rows in set (0.00 sec)

mysql> select * from Table_ limit 10;
+------------+--------------+---------------+-------------+
| sku        | numberOfLegs | numberOfSeats | shape       |
+------------+--------------+---------------+-------------+
| 0EKsH99oc8 |            8 |             6 | Round       |
| 1masnS2t5q |            7 |             4 | Rectangular |
| 2h36eG3k1r |            7 |             9 | Rectangular |
| 2IQOLQOPsC |            6 |             4 | Round       |
| 2k6FCNLgQa |            9 |             6 | Round       |
| 2orEiGE7st |            5 |            10 | Rectangular |
| 2sZn2Q9ZtV |            6 |             6 | Rectangular |
| 4arqe70auT |            4 |             9 | Round       |
| 4NeTvOLwsz |            7 |             8 | Rectangular |
| 4OsyaTUMoa |            9 |             5 | Round       |
+------------+--------------+---------------+-------------+
10 rows in set (0.00 sec)

mysql> select * from Desk limit 10;
+------------+-------+-----------------+
| sku        | angle | numberOfDrawers |
+------------+-------+-----------------+
| 0t5AqpGvtk |     7 |               6 |
| 0ZtpDsH7UG |     8 |               1 |
| 1Lnhya8ugI |    11 |               3 |
| 29ilUDyZOJ |    10 |               4 |
| 301ctu2OrC |    34 |               6 |
| 38JWLVLBbI |    14 |               6 |
| 3deDlxVPvv |     9 |               4 |
| 3XZONY6DCR |    39 |               3 |
| 5MugiPD2JZ |     2 |               1 |
| 5sX4ygONIN |    12 |               2 |
+------------+-------+-----------------+
10 rows in set (0.00 sec)

mysql> select * from Stool limit 10;
+------------+--------------+------------+-----------+
| sku        | numberOfLegs | hasCushion | hasSwivel |
+------------+--------------+------------+-----------+
| 0EbNx7hvlO |            3 |          1 |         1 |
| 0snow42N9H |            5 |          0 |         0 |
| 1cJboIO6HA |            4 |          1 |         0 |
| 1IemqAoqLk |            3 |          0 |         1 |
| 1ZIidFPuSJ |            4 |          1 |         0 |
```

```
| 2O8vTBpTSp |            4 |          1 |         1 |
| 3CyrMBvrG2 |            5 |          1 |         1 |
| 3qI7CVoPFi |            3 |          1 |         0 |
| 4mFkBSA9sZ |            3 |          0 |         1 |
| 52CLg2zVKj |            3 |          0 |         0 |
+------------+--------------+------------+-----------+
10 rows in set (0.00 sec)

mysql> select * from Cabinet limit 10;
+------------+----------------------+----------+
| sku        | numberOfCompartments | capacity |
+------------+----------------------+----------+
| 09S4mRvuGE |                    6 | 8        |
| 0eLPBgmxXc |                    3 | 6        |
| 0F6T8XT2R1 |                    1 | 4        |
| 17ApLkPoU6 |                    2 | 9        |
| 1Ddecxm7cb |                    1 | 7        |
| 1MeDZxrjEP |                    4 | 6        |
| 1ykxqSZlb0 |                    1 | 2        |
| 2YXJ88JhZg |                    2 | 3        |
| 3fPhRkO5BI |                    3 | 10       |
| 3jPPB5DgV3 |                    2 | 4        |
+------------+----------------------+----------+
10 rows in set (0.00 sec)

mysql> select * from Bedframe limit 10;
+------------+-----------------+--------+
| sku        | size_           | depth_ |
+------------+-----------------+--------+
| 0EbNx7hvlO | Twin XL         |     35 |
| 1xIdirGiRI | California King |     33 |
| 2oXo7jOoTO | California King |     26 |
| 4mFkBSA9sZ | King            |     41 |
| 4OsyaTUMoa | California King |     38 |
| 4SxmLVmwvL | King            |     24 |
| 4uEa0JgHI2 | Twin            |     25 |
| 4uSXR0shjD | Full            |     28 |
| 4V4wXGK9BV | California King |     31 |
| 53S4nBUolG | Queen           |     34 |
+------------+-----------------+--------+
10 rows in set (0.00 sec)

mysql> select * from features_Feature limit 10;
+-------------+--------------------+--------+
| modelNumber | description        | count_ |
```

```
+-------------+---------------------+--------+
| 00aWzohu8g  | Carved Inlays       |      5 |
| 00aWzohu8g  | Fancy Molding       |      3 |
| 00aWzohu8g  | Gold Hinges         |      2 |
| 00aWzohu8g  | Gold Inlaid Designs |      1 |
| 00aWzohu8g  | Ivory Handles       |      5 |
| 00aWzohu8g  | Padded Feet         |      1 |
| 00aWzohu8g  | Studded Corners     |      1 |
| 00aWzohu8g  | Textured            |      4 |
| 02SVZ28q47  | Carved Inlays       |      4 |
| 02SVZ28q47  | Claw Feet           |      2 |
+-------------+---------------------+--------+
10 rows in set (0.00 sec)
```

Group Work

Timothy: Wrote the bulk of the C++ data generator, including rewriting it several times as requirements changed.

Alexander: Wrote the bulk of the MySQL table creation statements, assisted with debugging the C++ data generator, and loaded the data into the database once it was generated.

Schuyler: Assisted with various parts of the table creation and data generation.