Database Design, CSCI 340, Spring 2016
2nd Exam, April 1

Multiple-Choice

1. Which of the following is equivalent to a "table"? (3 pts.)
    a. record
    b. relation
    c. relationship
    d. constraint
    e. schema

2. Which of the following is not part of the vocabulary of database keys? (3 pts.)
    a. Referential key
    b. Composite key
    c. Primary key
    d. Candidate key
    e. Foreign key

3. Denormalizing a relation _______ writes and ________ reads. (3 pts.)
    a. speeds up, speeds up
    b. slows down, slows down
    c. speeds up, slows down
    d. slows down, speeds up
    e. has no effect

4. Views: (3 pts.)
    a. Increase execution efficiency of queries
    b. Help enforce referential integrity
    c. Provide a consistent picture of the database even when the database changes
    d. None of the above
    e. All of the above

5. When you want to see all of the awards, even those not yet granted to a student, replace JOIN in the following

    Student JOIN Award ON Student.studentID=Award.studentID

    with: (3 pts.)
    a. LEFT OUTER JOIN
    b. RIGHT OUTER JOIN
    c. THETA JOIN
    d. EQUI JOIN
    e. JOIN (i.e. no change)

Short Answer

6. In reference to information stewardship, the term “data retention” refers to an organization retaining data past the time when the organization has performed the activity for which the data was collected. Discuss the consequences of data retention policies on privacy. (5 pts.)

   Retaining data longer than it is needed has several consequences relating to data stewardship.

   - Unauthorized people may access the information
   - The organization may be tempted to use the data in ways that the owner did not agree to
   - The information may become outdated. If it is kept, there need to be ways for the owner of the information to view and update their information

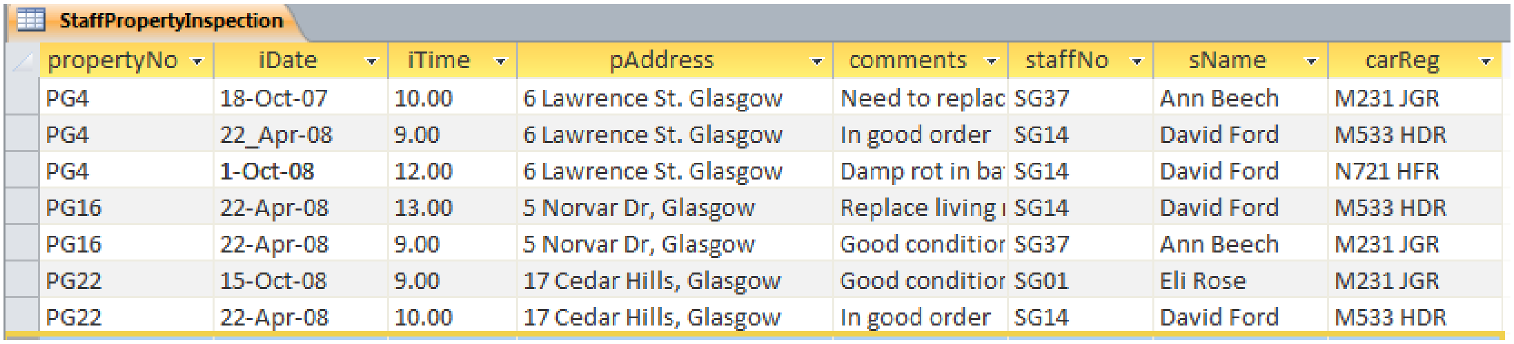
7. Explain why it is sometimes said that normalization is about design, while denormalization is about optimization. (5 pts.)

   Normalizing the relations in a database reduces anomalies in the database, so that insertions, deletions and updating records does not cause the database to loose information or have information become inconsistent. Thus normalization makes well designed databases.

   However, normalizing the relations in a database is done by decomposing the relations into multiple relations, whereby the original relation can be recreated by joining the decomposed relations together. Joins are expensive. In some cases it is desirable to pair fields which are frequently accessed together in the same table, thereby avoiding joins. Thus, to improve performance of the database (the time taken to get query results) some relations may be denormalized, repeating data even when it could cause anomalies.

Problem Solving

8. Consider the following relation.



propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg
PG4	18-Oct-07	10.00	6 Lawrence St. Glasgow	Need to replac	SG37	Ann Beech	M231 JGR
PG4	22_Apr-08	9.00	6 Lawrence St. Glasgow	In good order	SG14	David Ford	M533 HDR
PG4	1-Oct-08	12.00	6 Lawrence St. Glasgow	Damp rot in ba	SG14	David Ford	N721 HFR
PG16	22-Apr-08	13.00	5 Norvar Dr, Glasgow	Replace living	SG14	David Ford	M533 HDR
PG16	22-Apr-08	9.00	5 Norvar Dr, Glasgow	Good conditior	SG37	Ann Beech	M231 JGR
PG22	15-Oct-08	9.00	17 Cedar Hills, Glasgow	Good conditior	SG01	Eli Rose	M231 JGR
PG22	22-Apr-08	10.00	17 Cedar Hills, Glasgow	In good order	SG14	David Ford	M533 HDR

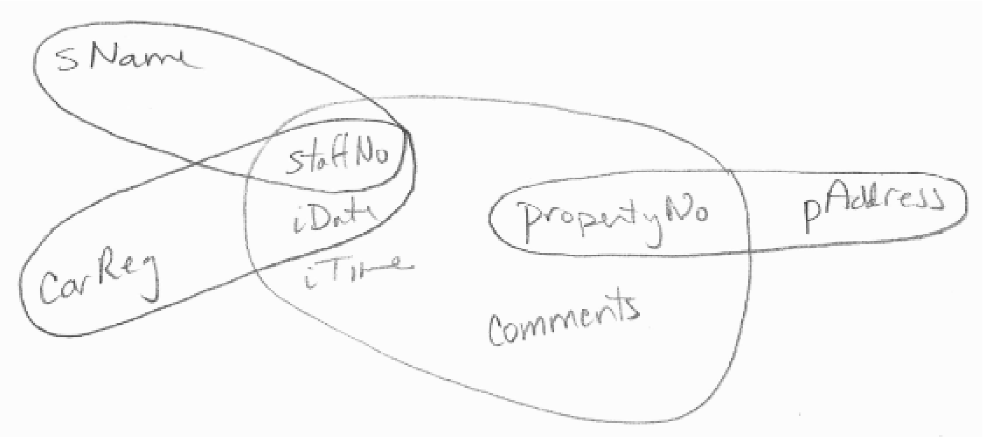
Say that you have been told that this relation has the functional dependencies:

propertyNo → pAddress
staffNo → sName
staffNo, iDate → carReg
staffNo, iDate, iTime → propertyNo, pAddress, comments, sName, carReg
propertyNo, iDate, iTime → staffNo, pAddress, comments, sName, carReg

a. Give all candidate keys for this relation. (5 pts.)

staffNo, iDate, iTime
propertyNo, iDate, iTime

b. Normalize the relation to a set of relations in BCNF if the relation is not already in BCNF. (10 pts.)



Relations:
Staff(staffNo, sName)
Only FD is staffNo → sName so relation is in BCNF.

Property(propertyNo, pAddress)
Only FD is propertyNo → pAddress so relation is in BCNF.

Car(staffNo iDate, carReg)
Only FD is staffNo, iDate → carReg so relation is in BCNF.

Inspection(staffNo iDate, iTime, propertyNo, comments)
FDs are:
    staffNo, iDate, iTime → propertyNo, comments
    propertyNo, iDate, iTime → staffNo, comments,
so relation is in BCNF.

Answer:
Staff(staffNo, sName)
Property(propertyNo, pAddress)
Car(staffNo iDate, carReg)
Inspection(staffNo iDate, iTime, propertyNo, comments)

9. The Friendly School Recruiting Office wants to record information about interactions that occur between someone interested in attending the school and recruiting office personnel. The recruiting office does not want to ask potential students lots of questions when they first inquire about the school, so they simply get the first and last name of the inquirer and assign that person a unique identification number. (The potential student never needs to be aware that the school has assigned them an identification number.) As more information about the potential student becomes available, such as the address (street, city, state, country, zip/country code), one or more phone numbers, and one or more email addresses, this information is recorded into the database. The system should be able to associate any number of phone numbers with the recruit and any number of email addresses. Identifiers such as “cell”, “home” or “preferred” should be able to be associated with each phone number. It should be possible to indicate which of the email addresses is preferred.

   A recruiter is assigned to every potential student. This recruiter has responsibility for doing follow up work, contacting the potential student, answering their questions, and keeping the student aware of what needs to be done to apply to the school. In some cases the recruiter assigned to a potential student changes. The database only keeps track of the current recruiter.

   An employee identification number, first and last name, and the telephone extension are kept for each recruiter. A recruiter may not have any students assigned to them, but typically he/she is assigned to several students.

   Recruiting personnel often talk with many potential students so they want the system to allow them to record notes about a potential student. Every note should be dated.If a new recruiter is assigned to work with the student, the new recruiter should be able to see the notes concerning the student.

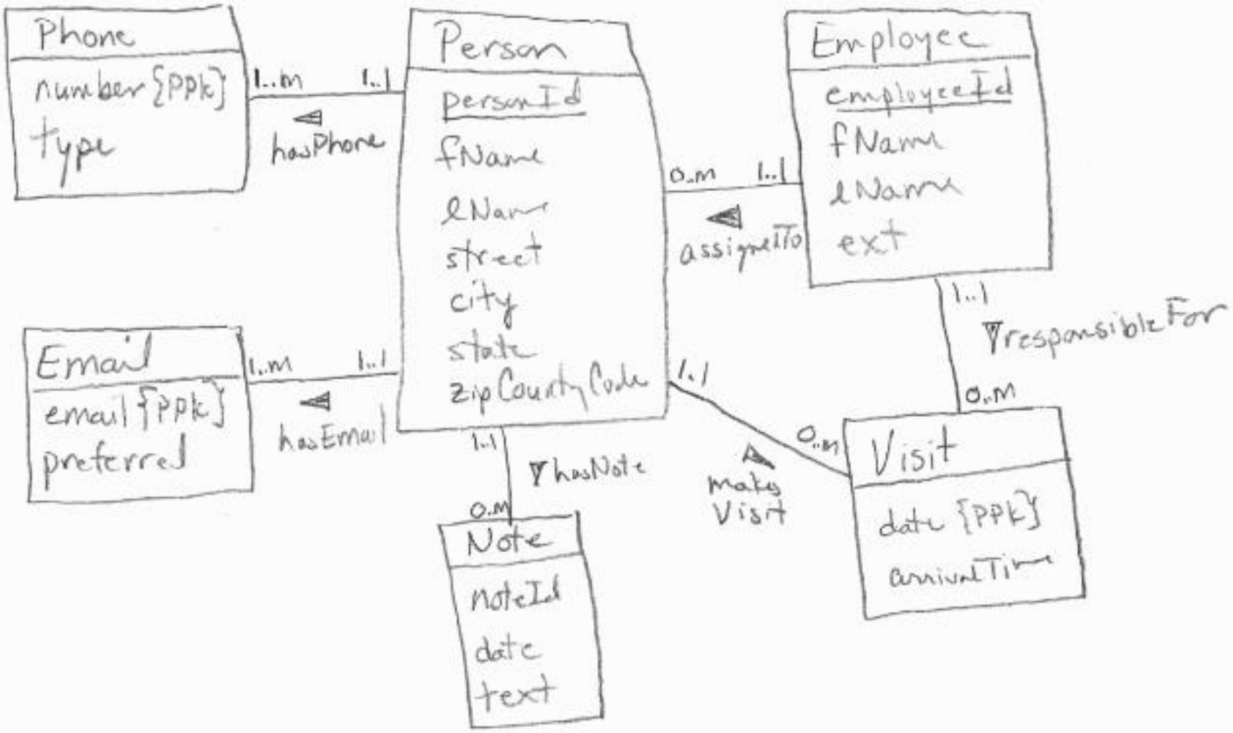
   Potential students may decide to make one or more visits to the school. One recruiter has responsibility for each site visit. This may or may not be the same recruiter who has responsibility for the student. The date and anticipated arrival time are kept for each visit. Students never make two site visits to the school on the same date.

   Imagine that you are responsible for developing this database application. Develop a conceptual model for it. This model should:

   - underline primary keys, (for partial primary keys use {PPK})
   - include relationships with minimum and maximum cardinalities
   - name the relationships indicating how to interpret the name via a triangular arrows ► and ◄.
   - multi-valued attributes are allowed in the design

(15 pts.)

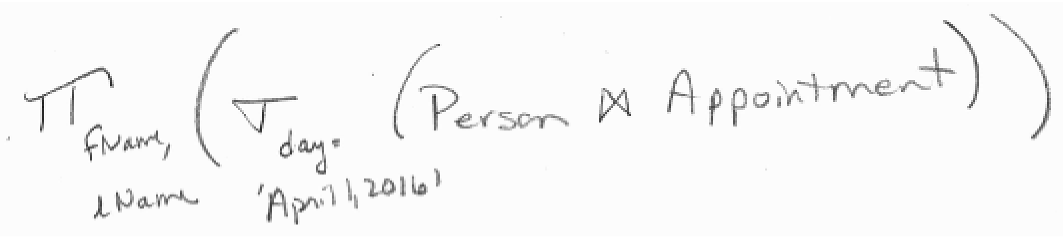
This page intentionally left blank☺



10. Consider the following database design:

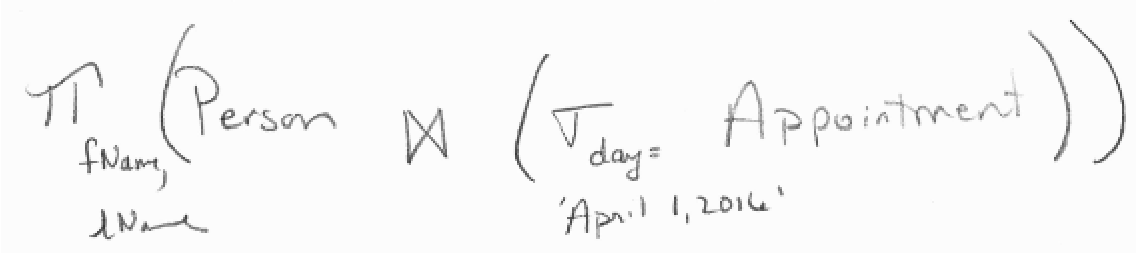
Person(ID, fName, lName)
Car(license, make, model)
Assignment(ID*, day*, license*)
Appointment(ID*, day, time)

a. Write a relational algebra statement which lists the first and last name of all people who have appointments on April 1, 2016. (5 pts.)



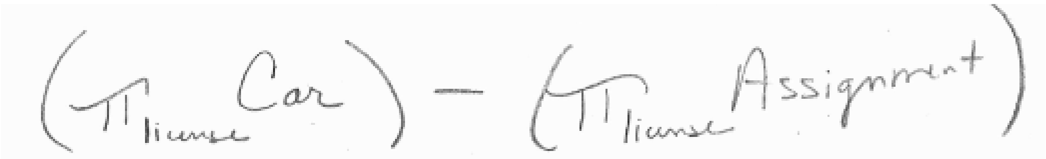
$$\pi_{fName,\ lName}\left(\sigma_{day=\text{'April 1, 2016'}}\left(Person \bowtie Appointment\right)\right)$$

or



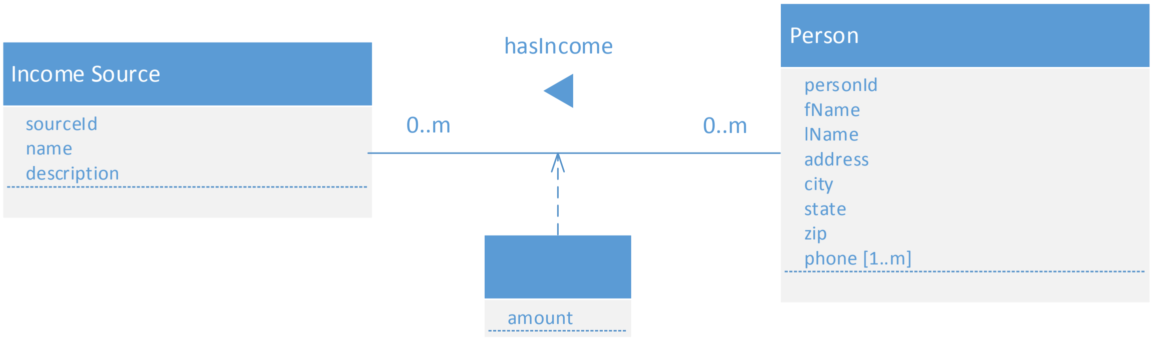
$$\pi_{fName,\ lName}\left(Person \bowtie \left(\sigma_{day=\text{'April 1, 2016'}}\ Appointment\right)\right)$$

b. Write a relational algebra statement which lists the license of every car which, according to the database, has never been assigned to a person. (5 pts.)

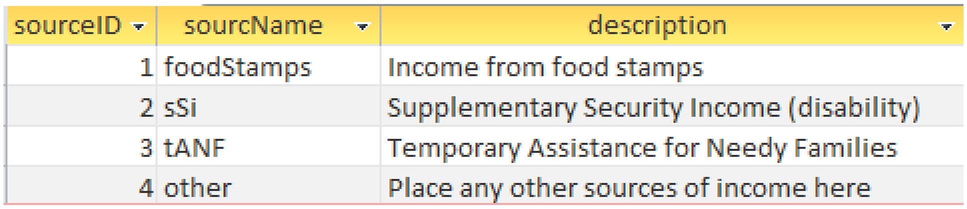


$$\left(\pi_{license}\ Car\right) - \left(\pi_{license}\ Assignment\right)$$

11. Butte Emergency Food Bank provides boxes of food to families whose income is under a certain level. For this reason, they need to track the income of clients. A conceptual model for tracking income could be the following.



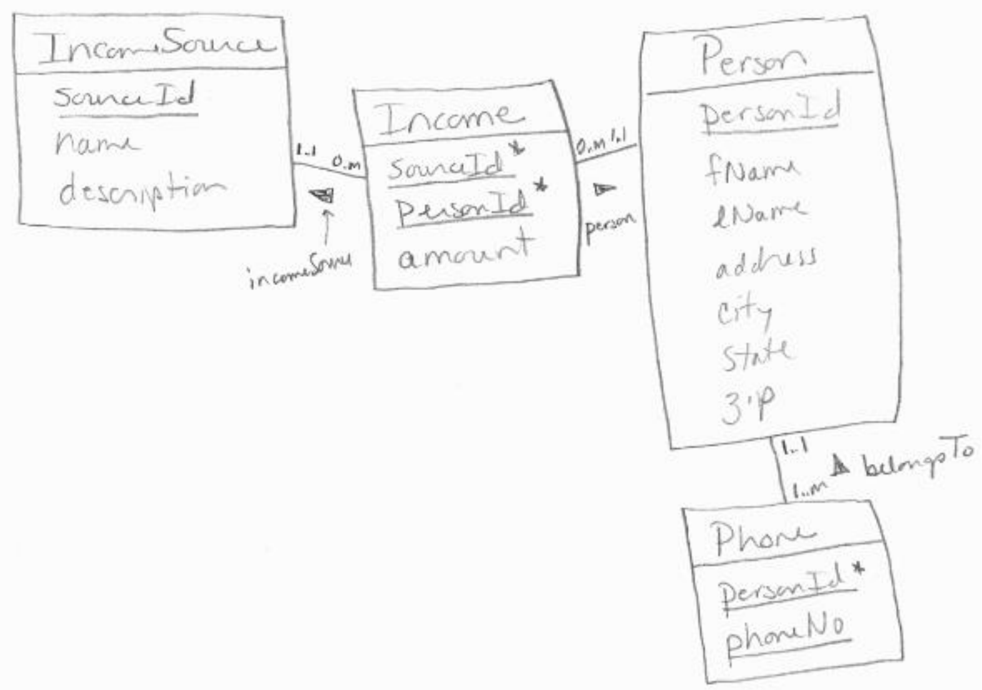
Income sources are:



sourceID	sourcName	description
1	foodStamps	Income from food stamps
2	sSi	Supplementary Security Income (disability)
3	tANF	Temporary Assistance for Needy Families
4	other	Place any other sources of income here

Recall that phone[1..m] is a multi-valued attribute which means that there can be one or more phone numbers associated with a person in the database.

Create a logical model for the conceptual models above. Include relationships, foreign keys, join tables, cardinalities and indicate key fields. (5 pts.)



12. Write the SQL statements to create the relations from your logical model in the previous question. Be sure to enforce referential integrity. (You can get full credit even if you don't remember the exact commands for specifying referential integrity.) (10 pts.)

```
CREATE TABLE IncomeSource
(sourceId INT PRIMARY KEY IDENTITY,
name CHAR(30),
description CHAR(100))


CREATE TABLE Person
(personId INT PRIMARY KEY IDENTITY,
fName CHAR(30),
lName CHAR(30),
address CHAR(50),
city CHAR(30),
state CHAR(2),
zip CHAR(30))

CREATE TABLE Phone
(phoneNo CHAR(11) PRIMARY KEY,
personId INT CONSTRAINT FK_Phone_Person FOREIGN KEY
REFERENCES Person(personId))

CREATE TABLE HasIncome
(personId INT CONSTRAINT FK_HasIncome_Person FOREIGN
KEY REFERENCES Person(personId),
  sourceId INT CONSTRAINT FK_HasIncome_Source FOREIGN
KEY REFERENCES IncomeSource(sourceId),
  amount DECIMAL(10,2))
```

13. Write statements to insert one record into each relation. (5 pts.)

```
INSERT INTO IncomeSource (name, description)
VALUES ('foodStamps', 'Income from food stamps')


INSERT INTO Person (fName, lName, address, city, state, zip)
 VALUES ('Eli', 'Rose', '1322 Flower Lane', 'Kahului', 'HI',
'96732')

INSERT INTO Phone (phoneNo, personId)
 VALUES ('12221212444', 1)


INSERT INTO HasIncome (personId, sourceId, amount)
 VALUES (1, 3, 254.66)
```

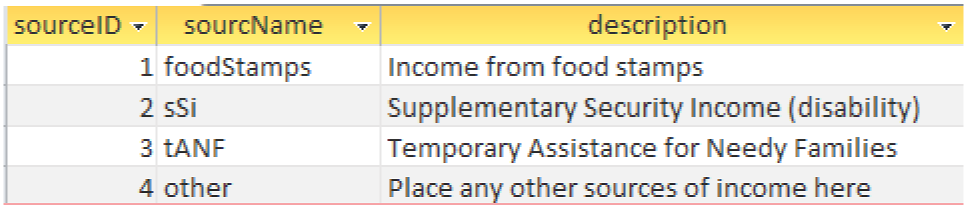
Database Design, CSCI 340, Spring 2016
2nd Exam, Computer Portion, April 1

Please turn in the first portion of the exam before beginning this portion. To complete this portion of the exam you may use your notes, any previous assignments, the text and/or the Internet. You may not communicate with anyone other than myself during this exam.

Consider the last question on the written portion of the exam:

Butte Emergency Food Bank provides boxes of food to families whose income is under a certain level. For this reason, they need to track the income of clients. A conceptual model for tracking income could be the following.

Income sources are:



sourceID	sourcName	description
1	foodStamps	Income from food stamps
2	sSi	Supplementary Security Income (disability)
3	tANF	Temporary Assistance for Needy Families
4	other	Place any other sources of income here

Recall that phone[1..m] is a multi-valued attribute which means that there can be one or more phone numbers associated with a person in the database.

Login to your SQLServer account and connect to your personal database.

14. Create the database which corresponds to the logical model which you created for the following conceptual model. (10 pts.)

```
CREATE TABLE IncomeSource
(sourceId INT PRIMARY KEY IDENTITY,
name CHAR(30),
description CHAR(100))


CREATE TABLE Person
(personId INT PRIMARY KEY IDENTITY,
fName CHAR(30),
lName CHAR(30),
address CHAR(50),
city CHAR(30),
state CHAR(2),
zip CHAR(30))

CREATE TABLE Phone
(phoneNo CHAR(11) PRIMARY KEY,
personId INT CONSTRAINT FK_Phone_Person FOREIGN KEY
REFERENCES Person(personId))

CREATE TABLE HasIncome
(personId INT CONSTRAINT FK_HasIncome_Person FOREIGN
KEY REFERENCES Person(personId),
  sourceId INT CONSTRAINT FK_HasIncome_Source FOREIGN
KEY REFERENCES IncomeSource(sourceId),
  amount DECIMAL(10,2))
```

15. Insert at least three records into each table. (5 pts.)

```
INSERT INTO IncomeSource (name, description)
VALUES ('foodStamps', 'Income from food stamps')

INSERT INTO IncomeSource (name, description)
VALUES ('sSI', 'Social Security Income (disailbity)')

INSERT INTO IncomeSource (name, description)
VALUES ('tANF', 'Temporary Assistance for Needy Families')
```

```
 INSERT INTO Person (fName, lName, address, city, state, zip)
 VALUES ('Eli', 'Rose', '1322 Flower Lane', 'Kahului', 'HI',
'96732')

 INSERT INTO Person (fName, lName, address, city, state, zip)
 VALUES ('Rio', 'Rose', '53 Lucky Lane', 'Lander', 'WY', '65521')

 INSERT INTO Person (fName, lName, address, city, state, zip)
 VALUES ('Jan', 'June', '555 Portolla Road', 'Atascadero', 'CA',
'93422')

 INSERT INTO Phone (phoneNo, personId)
 VALUES ('12221212444', 1)

 INSERT INTO Phone (phoneNo, personId)
 VALUES ('12221212445', 1)

 INSERT INTO Phone (phoneNo, personId)
 VALUES ('12221212446', 1)

 INSERT INTO HasIncome (personId, sourceId, amount)
 VALUES (1, 3, 254.66)

 INSERT INTO HasIncome (personId, sourceId, amount)
 VALUES (1, 2, 100.00)

 INSERT INTO HasIncome (personId, sourceId, amount)
 VALUES (3, 3, 259.66)
```