CSCE 310 Homework 3
Raymond Zhu
932008555

Problem 1:

(A) Functional dependency is a statement that describes the relation between two attributes in a table. The relationship is there when one attribute is dependent on the other attribute. Denoted such as A -> B where attribute B is dependent of A.

(B) A trivial functional dependency is when the right side (dependent attribute) is a subset of the other attribute

(C) These functional dependencies determine only part of a row because there are two partial dependencies
- ABC → D // D is dependent on C
- B → D // D is dependent on B
- A → C

(D) What makes a table not 3NF is that there are transitive dependencies. In the cases below, D → C is a transitive dependency on AB.
- AB → C
- AB → D
- D → C // This is transitive where AB → D → C

(E) Given two functional dependencies, A → C and B → D, it can be said that R(A, B, C, D) with key AB is in BCNF because the super key AB determined one or more of other attributes. In our case, the attributes are C & D.

(F) The relation can be decomposed into two relations R1 and R2 where
R1: S(A, B, C) with the functional dependency of FD: A → ABC
R2: T(C, D) with the functional dependency of FD: C → D
Both relations satisfy conditions of 3NF, there are no transitive dependencies

(G) Assuming the * means multiplication of the two attributes A and D from the relation, such a SQL statement can look like
- SELECT R1.A*R2.D FROM R1 INNER JOIN R2 ON R1.C = R2.C

Problem 2:

Given the schema S(A, B, C, D) with FD's A → B, B → C, and B → D

(A) What are all the nontrivial FD's that follow the from given FD's?

To calculate the FD's, we need to compute the closure of the attributes

For one attribute, we have:

- A+ = { A, B, C, D }
- B+ = { B, C, D }
- C+ = { C }
- D+ = { D }

For two attributes, we have:

- AB+ = { A, B, C, D }
- AC+ = { A, B, C, D }
- BC+ = { B, C, D }
- BD+ = { B, C, D }
- CD+ = { C, D }
- DA+ = { D, A, B, C }

For three attributes, we have:

- ABC+ = { A, B, C, D }
- ABD+ = { A, B, C, D }
- ACD+ = { A, B, C, D }
- BCD+ = { B, C, D }

And so, the FD's are

One Attributes:	A → C, A → D,
Two Attributes:	AB → D, AC → B, AC → D, BC → D, DA → B, DA → C,
Three Attributes:	ABC → D, ABD → C, ACD → B

(B) What are all the keys of R?

A, AB, AC, DA are all keys because the other functional dependencies do not contain the entire relation S(A, B, C, D) as its closure.

(C) A super key contains the keys listed above but cannot be a key. Since all keys share attribute A, all super keys need to contain attribute A. And so, all the super keys in relation S are ABC, ACD, ABD, ABCD.

Problem 3:

Given R(A,B,C,D,E) and FD's A → B, BC → E, DE → A

To start of this problem, the definition of a candidate key is a column or a set of columns which uniquely defines a table (the basis of a super key). Given the functional dependencies listed above, ABCDE, ABCD, and BDE are not candidate keys but rather super keys. This is because, the functional dependencies still exist even when one or more attributes are removed.

However, if we look at ACD, removing either attribute will make the functional dependencies listed above void. Because the FD's are preserved given this set of attributes as a super key, ACD is considered the candidate key since it is the basis of all super keys in the relation.

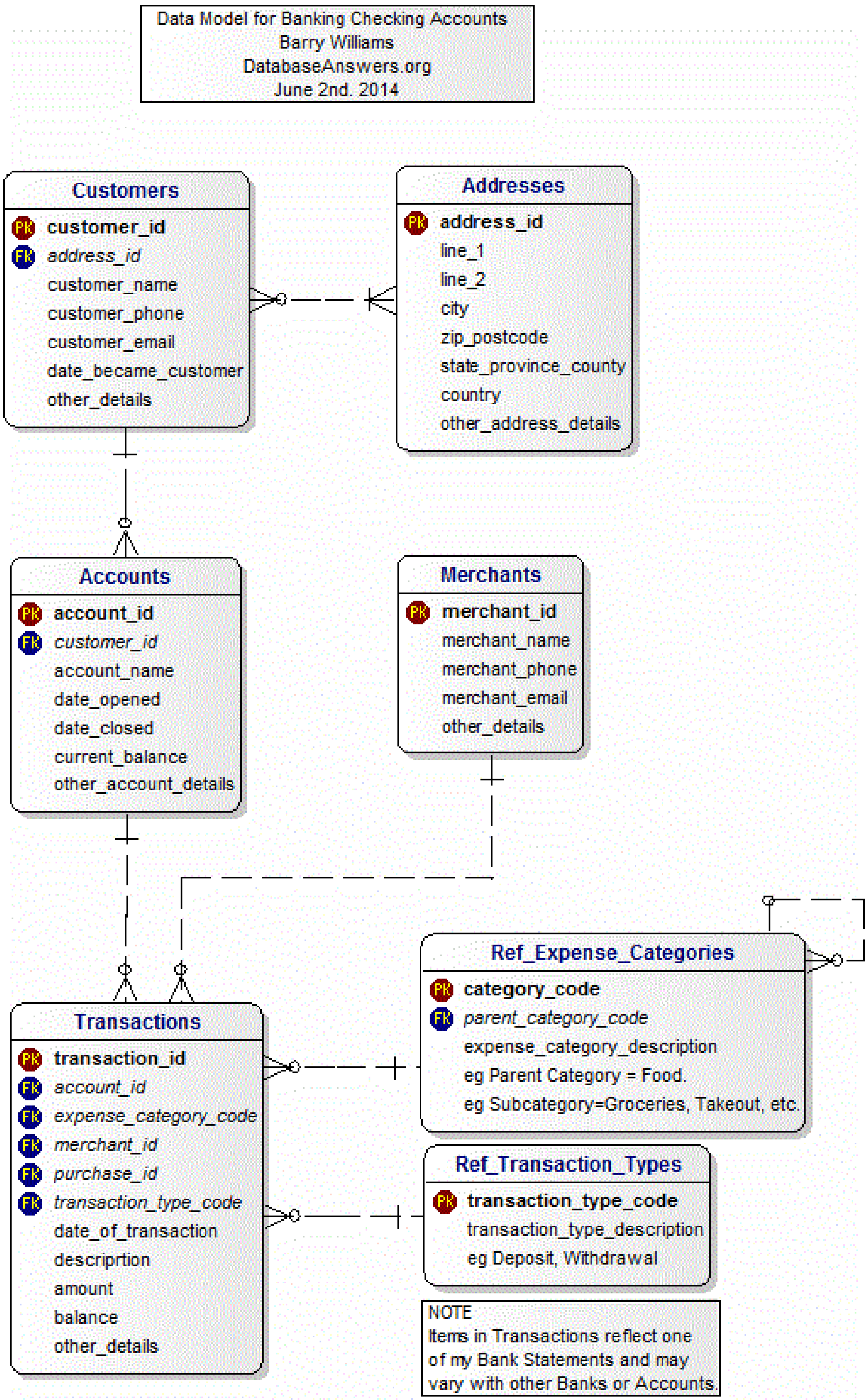
Problem 4:

(A) HS is the key for the relation Courses. They key contains all attributes in it's closure
(B) The functional dependencies for the relation are verifiable. Since the right side of every dependency contains only a single attribute, the functional dependency cannot be removed. In addition, neither can the attributes on left side of the dependency be altered, or else redundancy can occur and/or dependencies violated.
(C) If we decompose courses into separate relations to satisfy 3NF to remove any redundancies and transitive dependencies, we get

R1(C,T)	with FD: C → T
R2(H, R, C)	with FD: HR → C
R3(H, T, R)	with FD: HT → R
R4(H, S, R)	with FD: HS → R
R5(C, S, G)	with FD: CS → G

All are in BCNF. This is because, as verified above, each relation is a minimal basis and are part of a super key which contains one or more of the following attributes in the courses relation.

Problem 5:


Data Model for Banking Checking Accounts
Barry Williams
DatabaseAnswers.org
June 2nd. 2014
Customers
PK customer_id
FK address_id
customer_name
customer_phone
customer_email
date_became_customer
other_details
Addresses
PK address_id
line_1
line_2
city
zip_postcode
state_province_county
country
other_address_details
Accounts
PK account_id
FK customer_id
account_name
date_opened
date_closed
current_balance
other_account_details
Merchants
PK merchant_id
merchant_name
merchant_phone
merchant_email
other_details
Ref_Expense_Categories
PK category_code
FK parent_category_code
expense_category_description
eg Parent Category = Food.
eg Subcategory=Groceries, Takeout, etc.
Transactions
PK transaction_id
FK account_id
FK expense_category_code
FK merchant_id
FK purchase_id
FK transaction_type_code
date_of_transaction
descriprtion
amount
balance
other_details
Ref_Transaction_Types
PK transaction_type_code
transaction_type_description
eg Deposit, Withdrawal
NOTE
Items in Transactions reflect one
of my Bank Statements and may
vary with other Banks or Accounts.

Problem 6:

(A) CREATE TABLE Valid_Answers (

```
Valid_Answer_ID int NOT NULL,
Question_ID int NOT NULL,
Valid_Answer_Text varchar(255),
PRIMARY KEY(Valid_Answer_ID),
FOREIGN KEY(Question_ID) REFERENCES Questions(Question_ID)
);
```

(B) The professor would use several tables. These tables include the Ref_Type_of_Question, Questions, Valid_Answers, Questions_in_Exams, Exams, Ref_Subjects tables.

(C) The application would modify the Questions, Exams, and Valid_Answers tables.

(D) The application would query the Student, Student_Answers, Student_Assessments, and Valid_Answers to administer the exam.

(E) The application will update the Student_Answers and Student_Assessments tables while a student works through the exam

An aggie does not lie cheat, steal, or tolerate those who do