

CS 33007 Introduction to Database System Design, Spring 2020

Homework1

Possible points: 85

Deadline: February 2nd, Sunday, 11:59 PM

1. Discuss the following terms with simple examples. *[15 points]*
- (a) Transaction- several functions in a database application that perform a single logical function
 - (b) Atomic Value- a value that is treated as if it were indivisible
 - (c) Database schema- the visual representation of a database to define how data is organized and related
2. Considering the following schema of a bank database,

Branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)

- (a) Identify the candidate keys and the primary key of each relation. *[15 points]*

Branch candidate key:

branch_name; assuming that there could be more than one branch in a given city but each branch must have a unique name

Branch primary key: branch_name

Customer candidate key: customer_name, customer_street, customer_city; all three are needed as the candidate key because there can be more than one customer in a given household which means there can be more than one customer in a given city and customer_name is not guaranteed to be unique.

Customer primary key: customer_name, customer_street, customer_city; all three are needed as the candidate key because there can be more than one customer in a given household which means there can be more than one customer in a given city and customer_name is not guaranteed to be unique.

Loan candidate key: loan number; assuming that each loan must have a unique number associated with it that is unique company wide

Loan primary key: loan number; assuming that each loan must have a unique number associated with it that is unique company wide

Borrower candidate key: loan_number; assuming that a given customer can take out more than one loan at a given time

Borrower primary key: loan_number; assuming that a given customer can take out more than one loan at a given time

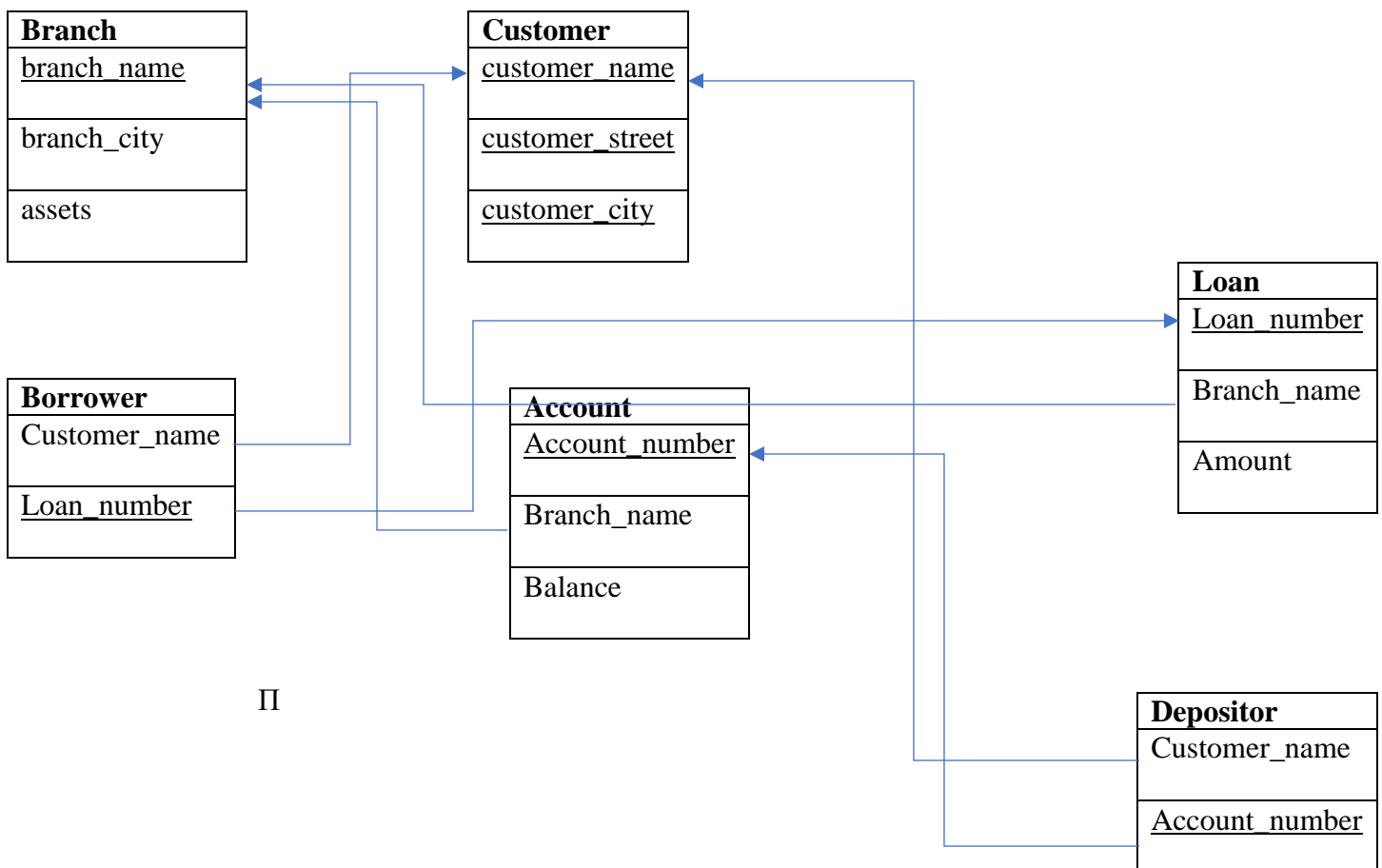
Account candidate key: account_number; assuming the balance can fluxuate and you can use any of the banks locations this must be the only candidate key

Account primary key: account_number; assuming the balance can fluxuate and you can use any of the banks locations

Depositor candidate key: account_number; assuming that a given customer can have more than one account with said bank

Depositor primary key: account_number; assuming that a given customer can have more than one account with said bank

(b) Draw a schema diagram for the database. Make sure to underline the primary keys. [15 points]



(c) Considering the schema above, write relational expressions for the following sentences [15 points]

- i. Find all customers from 'Kent' city.

$\sigma_{\text{customer_city}='Kent'}(\text{Customer})$

- ii. Find the names of all borrowers with a loan amount greater than \$5000

$\Pi_{\text{customer_name}} (\sigma_{\text{amount} > \$5000} (\text{Borrower} \bowtie \text{Loan}))$

- iii. Find the names of all depositors who have an account with a balance greater than \$9,000 at the “Kent” branch.

$\Pi_{\text{customer_name}} (\sigma_{\text{branch_name} = \text{“Kent”}} (\sigma_{\text{balance} > \$9000} (\text{Depositor} \bowtie \text{Account})))$

(d) What is **Cartesian-product**? Explain its importance in the relational data model. *[10 points]*

Cartesian-product is a way to join tables that doesn’t care about the relationship between said tables, in other words you can join tables that have nothing in common. It is important if you wish to combine two table and said tables don’t have any common attributes.

3. List superkeys and candidate keys of the following relation schemas of hospital database, *[15 points]*

Doctors (DoctorID, email, Name, MailingAddress, Salary)

Superkeys: DoctorID, email, Name, MailingAddress, Salary; DoctorID, email, Name, MailingAddress; DoctorID, email, Name, Salary; DoctorID, email, MailingAddress, Salary; DoctorID, Name, MailingAddress, Salary; DoctorID, email, Name; DoctorID, email, MailingAddress; DoctorID, email, Salary; DoctorID, Name, MailingAddress; DoctorID, Name, Salary; DoctorID, MailingAddress, Salary; DoctorID, email; DoctorID, Name; DoctorID, MailingAddress; DoctorID, Salary; DoctorID; email, Name, MailingAddress, Salary; email, Name, MailingAddress; email, Name, Salary; email, MailingAddress, Salary; email, Name; email, MailingAddress; email, Salary; email

Candidate Keys: DoctorID; email;

Patient (PatientID, SSN, name, address)

Superkeys: PatientID, SSN, name, address; PatientID, SSN, name; PatientID, SSN, address; PatientID, name, address; PatientID, SSN; PatientID, name; PatientID, address; PatientID; SSN, name, address; SSN, name; SSN, address; SSN

Candidate Keys: PatientID; SSN

VisitingRecords(PatientID, DoctorID, dateOfVisit)

Superkeys: PatientID, DoctorID, dateOfVisit; PatientID, dateOfVisit

Candidate Keys: PatientID, dateOfVisit