CSE 5330 – Database Systems Sec 002

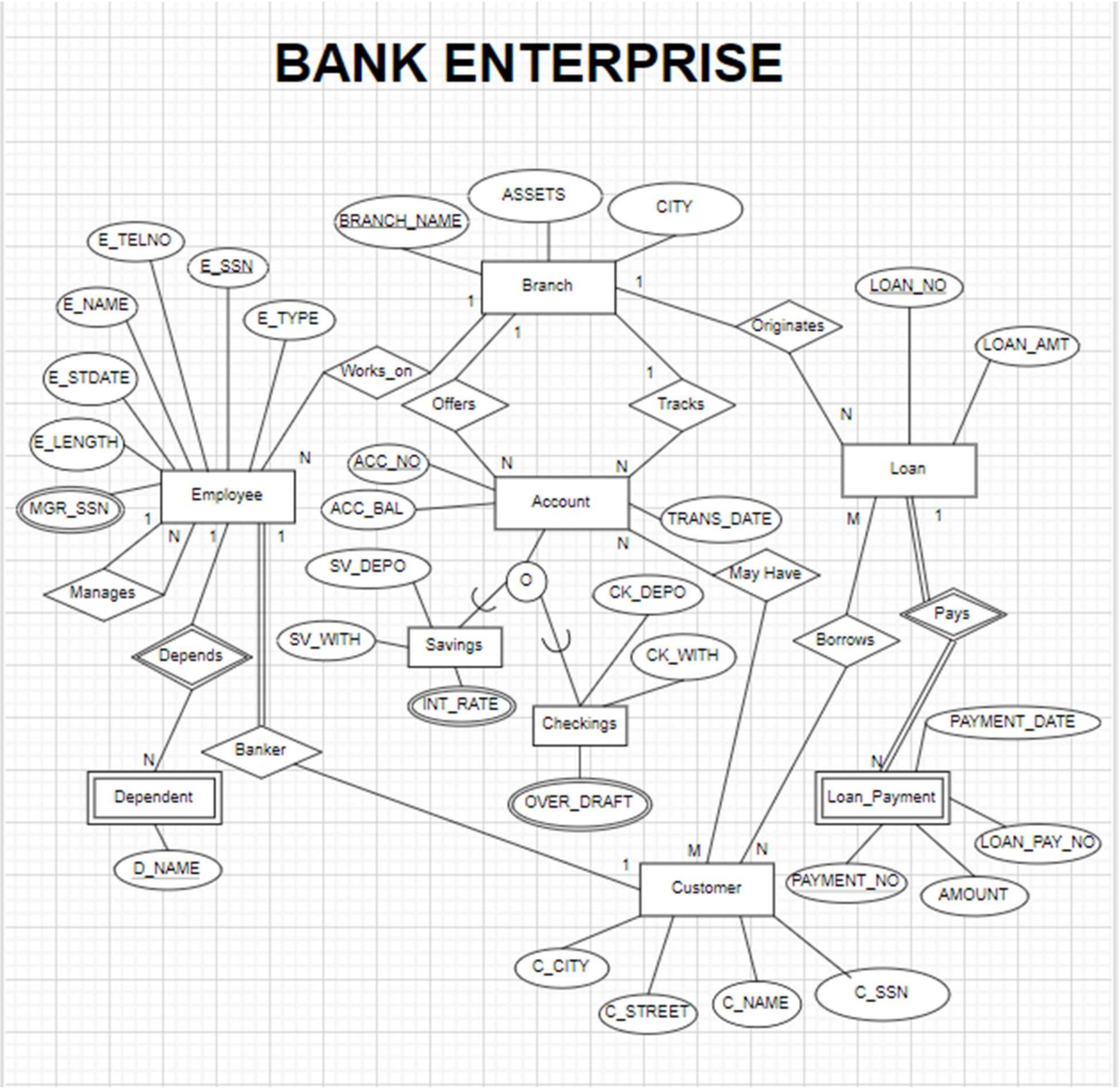
Project 2

Submitted By:

1) Rajat Singh Bahadur (1001878700)

2) Riddhi Baburao Joshi (1001955863)

1.a) Extended Entity Relationship (EER) Schema Diagram:



Rajat Singh Bahadur (1001878700)
Riddhi Baburao Joshi (1001955863)

Design of Choices:

For Project 2 we have created a Bank Enterprise database as per below design considerations:

- Branch is the initial entity of the Bank Enterprise database.
- Branch entity has attributes like Branch_Name as primary key, Assets, and City.
- Branch offers Account which is another entity having Acc_No as the primary key and other attributes like Acc_Bal, Trans_Date, Account_type, Branch_name being referenced from Branch.
- These Accounts are divided into two types which are Savings and Checkings. It has Acc_No as the referential key. Savings has the following attributes - SV_Depo, Sv_With, and Int_Rate. Checkings has Sv_Depo, Sv_With, and Over_draft as its attributes.
- Employee and Branch entities are related with many to one relation respectively. Employee has attributes E_Ssn (primary key), E_Telno, E_Name, E_Stdate, E_Length, Mgr_Ssn, Branch_name,E_Type. Mgr_Ssn is taken as multivalued attributes since several employees can have one manager.
- Dependent is a weak entity of Employee entity since Dependent table does not have a primary key, D_Name is considered as partial key. E_Ssn is the foreign key for Dependent entity referring from Employee.
- Employees can act as loan officer or personal banker for a particular customer.
- Customers can have accounts or can borrow loans.
- Customer has many too many relationships with Account and Loan entity.
- Customer entity provides the data related to the customer's Name, City, Street, and SSN and E_SSN where SSN is the primary key and E_SSN is the foreign key referenced from Employee.
- Loan originates from the branch. The loan entity has Loan_No as its primary key, Branch_name and Loan_amt. Loan_No is referenced by Loan_Payments, a weak entity.
- Loan_Payment keeps track of the payments made by every customer by storing the payment_date, payment_no, amount and loan_pay_no.
- For the relation – “Borrows” we have an entity called cust_loan providing the details about the loan taken by customers.
- For the relation – “May Have” we have an entity called cust_account providing the details about customers accounts.
- Cust_loan and cust_account are designed to satisfy the requirements of customer having one or more accounts or loan and vice versa.

Assumptions:

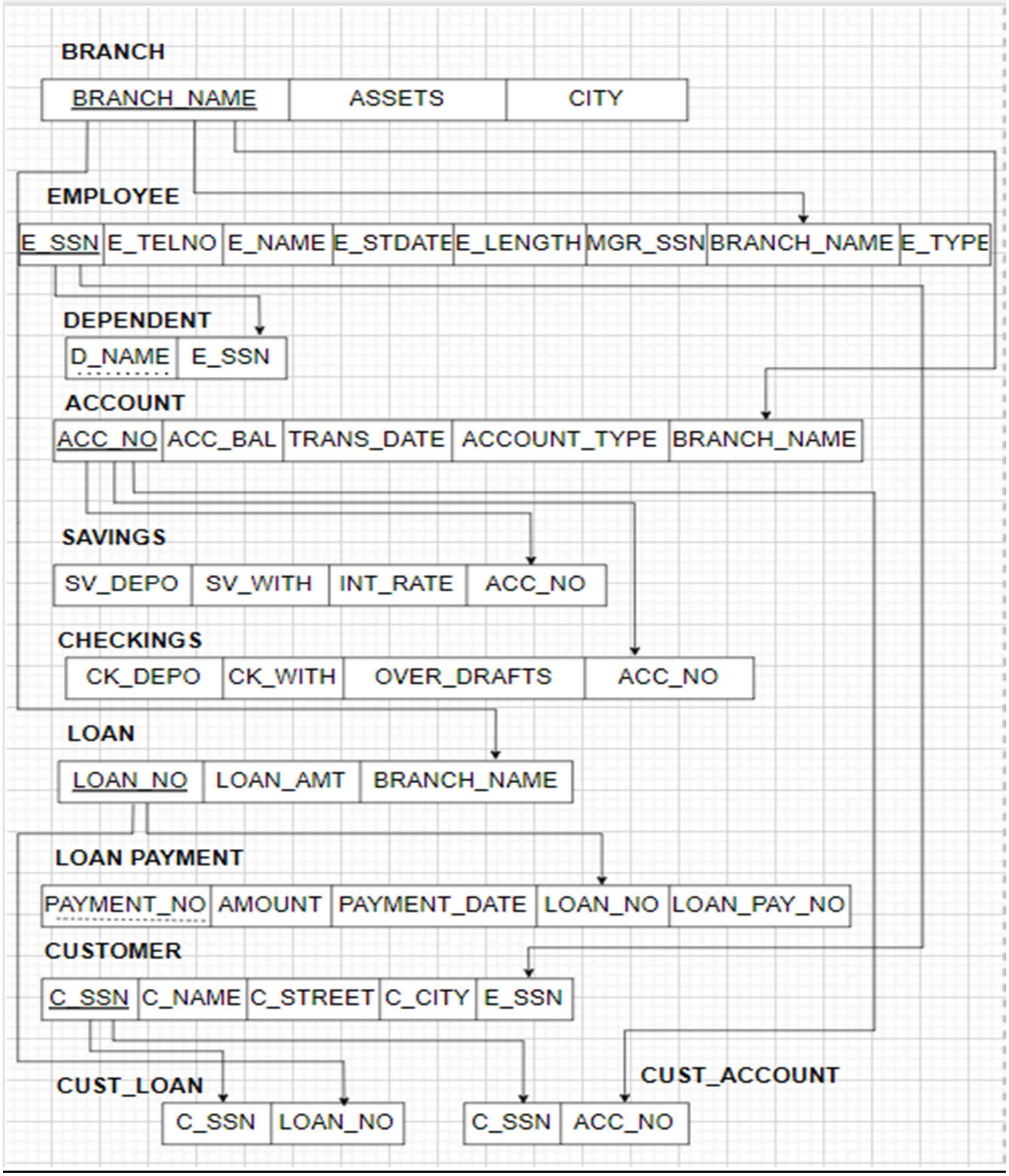
- Instead of using Bank as an entity, we have used Branch as an initial entity to create the database since there were no significant attributes mentioned for creating Bank.
- Considering the fact that each branch is located in a particular city, the City attribute cannot be multivalued.
- Every manager is an employee of the bank.
- A separate entity for Dependent is created considering that an employee can have more than one dependent and cannot be entered into the Employee entity.

- Savings and checkings accounts are overlapping since a customer can have both savings and checking accounts.

Limitations:

- Design shows that a city can have only one branch of the same bank.

1.b) EER to Relational Schema Mapping:



BRANCH

BRANCH_NAME	ASSETS	CITY

EMPLOYEE

E_SSN	E_TELNO	E_NAME	E_STDATE	E_LENGTH	MGR_SSN	BRANCH_NAME	E_TYPE

DEPENDENT

D_NAME	E_SSN

ACCOUNT

ACC_NO	ACC_BAL	TRANS_DATE	ACCOUNT_TYPE	BRANCH_NAME

SAVINGS

SV_DEPO	SV_WITH	INT_RATE	ACC_NO

CHECKINGS

CK_DEPO	CK_WITH	OVER_DRAFTS	ACC_NO

LOAN

LOAN_NO	LOAN_AMT	BRANCH_NAME

LOAN PAYMENT

PAYMENT_NO	AMOUNT	PAYMENT_DATE	LOAN_NO	LOAN_PAY_NO

CUSTOMER

C_SSN	C_NAME	C_STREET	C_CITY	E_SSN

CUST_LOAN

C_SSN	LOAN_NO

CUST_ACCOUNT

C_SSN	ACC_NO

Create Table Statements:

BRANCH:

```
CREATE TABLE BRANCH (BRANCH_NAME VARCHAR(30),
ASSETS VARCHAR(30),
CITY VARCHAR(20),
PRIMARY KEY(BRANCH_NAME));
```

EMPLOYEE:

```
CREATE TABLE EMPLOYEE (
E_SSN VARCHAR(12),
E_TELNO INT,
E_NAME VARCHAR(20),
E_STDATE VARCHAR(20),
E_LENGTH INT,
MGR_SSN VARCHAR(12),
BRANCH_NAME VARCHAR(30),
E_TYPE VARCHAR(30),
PRIMARY KEY(E_SSN),
FOREIGN KEY(BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));
```

DEPENDENT:

```
CREATE TABLE DEPENDENT (
D_NAME VARCHAR(20),
E_SSN VARCHAR(12),
PRIMARY KEY(D_NAME),
FOREIGN KEY (E_SSN) REFERENCES EMPLOYEE(E_SSN));
```

ACCOUNT:

```
CREATE TABLE ACCOUNT(
ACC_NO INT,
ACC_BAL FLOAT,
TRANS_DATE VARCHAR(10),
ACCOUNT_TYPE VARCHAR(20),
BRANCH_NAME VARCHAR(30),
PRIMARY KEY(ACC_NO),
FOREIGN KEY (BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));
```

LOAN:

```
CREATE TABLE LOAN(
LOAN_NO INT,
LOAN_AMT FLOAT,
BRANCH_NAME VARCHAR(20),
PRIMARY KEY (LOAN_NO),
FOREIGN KEY (BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));
```

CUSTOMER:

```
CREATE TABLE CUSTOMER (
C_SSN VARCHAR(12),
C_NAME VARCHAR(20),
C_STREET VARCHAR(20),
C_CITY VARCHAR(20),
E_SSN VARCHAR(12),
PRIMARY KEY (C_SSN),
FOREIGN KEY (E_SSN) REFERENCES EMPLOYEE(E_SSN));
```

LOAN_PAYMENT:

```
CREATE TABLE LOAN_PAYMENT(

PAYMENT_NO INT,

AMOUNT FLOAT,

PAYMENT_DATE VARCHAR(10),

LOAN_NO INT,

LOAN_PAY_NO VARCHAR(6),

PRIMARY KEY(PAYMENT_NO),

FOREIGN KEY (LOAN_NO) REFERENCES LOAN(LOAN_NO));
```

SAVINGS:

```
CREATE TABLE SAVINGS(

SV_DEPO FLOAT,

SV_WITH FLOAT,

INT_RATE FLOAT,

ACC_NO INT,

FOREIGN KEY (ACC_NO) REFERENCES ACCOUNT(ACC_NO));
```

CHECKINGS:

```
CREATE TABLE CHECKINGS(

CK_DEPO FLOAT,

CK_WITH FLOAT,

OVERDRAFTS FLOAT,

ACC_NO INT,

FOREIGN KEY (ACC_NO) REFERENCES ACCOUNT(ACC_NO));
```

CUST_LOAN:

```
CREATE TABLE CUST_LOAN(
C_SSN VARCHAR(12),

LOAN_NO INT,

FOREIGN KEY (C_SSN) REFERENCES CUSTOMER(C_SSN),

FOREIGN KEY (LOAN_NO) REFERENCES LOAN(LOAN_NO));
```

CUST_ACCOUNT:

```
CREATE TABLE CUST_ACCOUNT(

C_SSN VARCHAR(12),

ACC_NO INT,

FOREIGN KEY (C_SSN) REFERENCES CUSTOMER(C_SSN),

FOREIGN KEY (ACC_NO) REFERENCES ACCOUNT(ACC_NO));
```