Part I. ER Modelling (50 points)

- Submit a diagram drawn by a drawing program (any program you like, you can use those **ER diagramming tools** from http://en.wikipedia.org/wiki/Entity-relationship_model), **NOT** hand drawn.
- If you feel unsure about some of your design choices, you can add explanations.
- Upload your diagram and optional explanations in a single pdf file called **pdf**.

Draw a single ER diagram that represents the specifications listed below.

1. A banking enterprise needs to store information about **Customers** (identified by *cid*, with *cname* attributes) and **Accounts** (identified by *aid*, with an overdraft limit *amount*). (8 points)
2. A customer is either a company with its *street*, *city*, or an individual with his/her gender, age. (10 points)
3. Customers own accounts. For each account that a customer owns, we want to store a start *date*, at which the account was opened, and a *pin* number that gives the customer access to the account. A customer can have multiple accounts, but an account can only be owned by one customer. (7 points)
4. Let us add three more entities. A **loan** has the following attributes: loan number, loan type, and amount. Each loan has a unique loan number. The attributes of a loan **payment** are date, amount and payment number. For a given loan, the payment number identifies a unique payment, but payments for different loans may share the same number. For example, payment #1 for Jack Smith identifies a unique payment in the amount of $100, but payment #1 for Jackie Chan may be a different payment in the amount of $1,000. A **branch** has the attributes, branch number, city, and street. Each branch has a unique branch number. (15 points)
5. Each loan is taken at a single unique branch. Customers borrow loans; a customer may have more than one loan, and a given loan may be associated with more than one customer (e.g., a couple may be co-signers on a mortgage). (8 points)
6. Presentation quality including submission file formats and diagram presentation. (2 points)

Note: If you feel that not all the details that you need to know have been completely specified, use your common sense. If in doubt, explain explicitly what assumptions you make and how they shape your model.

Part II: Creating Relational Schemas in SQL for Part I (50 points)

- Please install DB Browser for SQLite on your computer and use it to run all your SQL statements. We use SQLite because it is free and probably the best lightweight database.
- Submit a single SQL script that contains all your SQL statements. This should be executable in SQLite. Name the file part2.sql .
- You may add optional explanations (e.g., why some constraints cannot be represented in SQL in your opinion). Please write these to another explanation_part2.pdf file.

Write SQL statements that translate your ER model for Part I into the relational model.

1. **Create a Relational Schema.** Create one table for each of the relations required to represent this banking enterprise. You may want to first translate the ER model into a relational schema, then the relational schema into SQL commands. E.g. in Chapter 3.1. the Students table is defined by "Students (sid: string, name: string, login: string, age: integer, gpa: real)".*You are not required to write down the relational schema in this format*. We will not grade you on the schema, only on the SQL statements for creating the tables. We just think the schema will help you. (20 points)
2. **Impose Integrity Constraints**. Impose the constraints implied or stated, such as domains (e.g., integer for *dno*), key constraints, foreign key constraints, and participation constraints (using NOT NULL). *Be sure to incorporate as many constraints mentioned in part I as possible.* If there is a constraint mentioned in part I that you cannot capture in SQL statements, describe what the constraint is and explain why it cannot be captured. (25 points)
3. **Presentation Quality.** (5 points)