You have been asked to design a preliminary database for a company called AAAAA Tech, Inc.

**Question #1** (60-point). Draw an ER diagram for the following situation (40-point) and state any assumptions you believe you have to make in order to develop a complete diagram (20-point):

AAAAA Tech, Inc., has 10 departments and about 4000 employees. The company needs an operational database to keep track of all employees, departments, and projects. Every employee has an employee number, social security number, employed year, current name, and other names used. Each employee has a certain position in a department, and may have dependents. For each dependent, a dependent name and age are stored in the database. There are two types of employees: marketing employees and engineering employees. Each marketing employee has a list of products. Each engineering employee works in one or more research areas, and the company has defined a set of research areas. Each department has a unique department number, a department name, and an address. Each employee can be involved in up to 3 projects, and a project is carried out by up to 10 employees. Each project must have one project head (i.e., an employee), a unique project number, and a project name. There are several other companies who supply various parts for the projects. AAAAA Tech wants to keep track of these suppliers, the parts they supply, and their prices. Each supplier is represented by a unique supplier number, supplier name, and address. A part can be a component of another part. For each part, a unique part number and a part name are stored. A project may need various parts and a part can be used in multiple projects.

**Question #2** (40-point). Convert your ER diagram into a relational database schema. Give the primary key, all candidate key(s), and all foreign key(s) of each relation schema. Note that, for this assignment, we don't consider integrity constraints, database normalization, and queries. Naming convention: use capital letters to write a relational schema name, and use lowercase letters to write a relational instance name. For example, “ACCOUNT” is a relational schema, “account” is a relational instance, and “account (ACCOUNT)” is a relation. Rename is allowed when it’s necessary.

Example:

account (ACCOUNT):
ACCOUNT = <acc_num, balance, branch>
pk = <acc_num>
ck=<acc_num>
fk=Ø

OR

account(ACCOUNT):
ACCOUNT = {acc_num, balance, branch}
pk = {acc_num}
ck={acc_num}
fk=Ø