Answer questions 1-4 based on the following definitions. Note that the relation **employee** has four attributes, where ID represents the employee identifier. Besides, the relation **works** describes the company which an employee (ID) works for, the corresponding salary, and the boss of the company.

```
create table employee
(ID      char(5),
name    char(5),
street    varchar(10),
city     varchar(10),
primary key (ID));
```

```
create table works
(ID          char(5),
company      varchar(10) ,
salary       integer,
boss         char(5),
primary key (ID, company),
foreign key (ID) references
 employee on update cascade);
```

```
create view tempview as
    select *
    from employee natural inner
         join works
    where company=  'Acer' ;
```

1. (10%) Consider the table "works". Please describe what constraints the **primary key** definition and the **foreign key** definition pose on the data.

2. (20%) Write an SQL expression for each of the following questions, respectively.
   (1) Output the ID's of those employees who live in "Taipei" and also output the companies which they work for.
   (2) Find the ID's of those employees who work for more than two companies.
   (3) Output the average salary of each company.
   (4) Using the view "tempview", output the person who earns the most salary at the company "Acer".

3. (10%) Suppose the following three function dependencies are satisfied by the relation "works".
   {ID company -> salary, ID company-> boss, company-> boss}
   (1) Please determine if the relation "works" is in BCNF.
   (2) Please describe the benefit of performing normalization in general.

4. (10%) Suppose there is a B^+-tree index on the attribute ID of the relation "employee", which is called the "i-index". Please give an efficient query evaluation plan to process the following query based on the index "i-index":
   "Output the names of those employees who work for the company "Asus".
   You need to explain why your answer is more efficient than other alternatives.

5. (20%) Draw an ER diagram to design the database satisfying all the following descriptions and constraints:
   - Record the information about each vehicle, such as its tag and model, etc.
   - Record the information of each owner, such as his/her ID and address, etc.
   - Each vehicle can have only one owner, but a person can own several vehicles.
   - There are at least two kinds of vehicles: cars and motorcycles. For a car, we want to record its type; for a motorcycle, we want to record its horsepower.

   For each entity set, design appropriate attributes and primary keys for them by yourselves. You can also assign attributes to relationship sets.

6. (10%) Construct a B^+-tree for the following set of key values:

   (1, 3, 5, 7, 9, 11, 13, 15, 17, 19)

   Assume that the tree is initially empty, values are added in ascending order, and the number of pointers that fit in one node is "**six**". (That is, the order is 6.) Construct the tree step by step and show the final tree.

7. (10%) The traditional relational database system usually wishes to maintain **four** important properties for any transaction to preserve the integrity of data. Please list them and explain their meanings.

8. (10%) Key-value stores are commonly used to represent big data. Please list **three** characteristics usually found in key-value stores, but not in traditional relational database systems.