Database design problems with solutions

Problem 1

Design an Entity-Relationship schema for a database of research projects. The database should contain the information about:

– projects: name, manager, budget, duration (in years), funding agency;

– employees: SSN, name, projects, salary;

– funding agencies: name, address.

Each project is funded by a single agency. Project names are unique within an agency. An employee can be associated with several projects. Managers are employees. *You can make any other additional assumptions that make sense in the real world.*
Solution.
The E-R schema:

– entity types:

   – Project (weak): attributes: Name (partial key), Budget, Duration, borrowed key Aname from Agency;
   – Employee: SSN (key), Name, Salary;
   – Manager;
   – Agency: Aname (key), Address.

– relationship types:

   – Manages(Manager,Project): 1:N;
   – Funding(Agency,Project): 1:N;
   – Employ(Project,Employee): N:M.

– **isa** relationships:

   – Manager **isa** Employee.

Problem 2

Produce a relational schema in BCNF from the E-R schema obtained in Problem 1. *If your approach is* guaranteed *to produce a schema in BCNF, then you do not have to* check *whether that condition holds.* Identify the keys and foreign keys. Eliminate redundancies.
Solution.
The relational schema:

– PROJECT(<u>Aname,Pname</u>,Budget,Duration), foreign key Aname references AGENCY(Aname);

– EMPLOYEE(<u>SSN</u>,Ename,Salary);

– MANAGER(<u>SSN</u>), foreign key SSN references EMPLOYEE(SSN);

- AGENCY(<u>Aname</u>,Address);

- MANAGES(SSN,<u>Aname,Pname</u>); foreign key (Aname,Pname) references PROJECT(Aname,Pname);

- EMPLOY(<u>Aname,Pname,SSN</u>), foreign keys: (Aname,Pname) references PROJECT(Aname,Pname) and SSN references EMPLOYEE(SSN).

PROJECT and MANAGES can be merged to yield PROJECT(<u>Aname,Pname</u>,Budget,Duration,Manager).

Problem 3

Let $R(ABC)$ be a relation schema together with the set of dependencies $F = \{C \rightarrow A, B \rightarrow A\}$.

1. Find the keys of R and check whether R is in BCNF or 3NF. Prove the answers, using the appropriate definitions.

2. If R is not in BCNF, provide a lossless join decomposition of R into BCNF and check whether it preserves dependencies.

Solution.

One key BC. Show: $BC \rightarrow ABC \in F^+$ (using Armstrong and derived axioms), $B \rightarrow ABC \notin F^+$ (using counterexample), $C \rightarrow ABC \notin F^+$ (using counterexample). One counterexample proof is enough.

R is not in BCNF (C and B does not contain a key) or 3NF (A is not a part of any key).

Possible lossless join decompositions (each is acceptable):

- $\{BC, AC\}$ (loses $B \rightarrow A$);

- $\{BC, AB\}$ (loses $C \rightarrow A$);

- $\{BC, AB, AC\}$ (preserves entire F).

  The decomposition $\{AB, AC\}$ *is not lossless join*.