A relational schema

```
EMPLOYEE(enumber, first-name, last-name, project-title,
                                           budget, deadline)
```

to eliminate redundancies, must be decomposed into the following relational schemas:

```
EMPLOYEE(enumber, first-name, last-name)
primary key = (enumber)

PROJECT(project-title, budget, deadline)
primary key = (project-title)

WORKS-ON(enumber, project-title)
primary key =(enumber, project-title)
foreign key 1 = (enumber) references EMPLOYEE(enumber)
foreign key 2 = (project-title) references
                                    PROJECT(project-title)
```

A relational schema
`MANAGER(enumber, first-name, last-name, mnumber)`

does not need to be decomposed.

```
MANAGER(enumber, first-name, last-name, mnumber)
primary key = (enumber)
foreign key = (mnumber) references MANAGER(enumber)
```

The relational schemas `EMPLOYEE` and `MANAGER` can be synthesised into one schema

```
EMPLOYEE-MANGER(enumber, first-name, last-name, mnumber)
primary key = (enumber)
foreign key = (mnumber) references
                                EMPLOYEE-MANAGER(enumber)
```

A relational schema

```
BUILDING(bnumber, bname, rnumber, area, enumber)
```

to eliminate redundancies, must be decomposed into the following relational schemas:

```
BUILDING(bnumber, bname)
primary key (bnumber)
candidate key (bname)
```

```
ROOM(bnumber, rnumber, area)
primary key = (bnumber, rnumber)
foreign key = (bnumber) references BUILDING(bnumber)

EMPLOC(enumber, bnumber, rnumber)
primary key =(enumber)
foreign key = (bnumber, rnumber) references ROOM(bnumber,
rnumber)
```

The relational schemas_EMPLOYEE-MANGER and EMPLOC can be synthesised into a relational schema:

```
EMPLOYEE-MANAGER-LOCATION(enumber, first-name, last-name,
mnumber, bnumber, rnumber)
primary key =(enumber)
foreign key 1 = (bnumber, rnumber) references ROOM(bnumber,
rnumber)
foreign key 2 = (mnumber) references
                         EMPLOYEE-MANAGER-LOCATION(enumber)
```

A relational schema

```
EQUIPMENT(serialnum, bnumber, rnumber)
```

does not need to be decomposed.

```
EQUIPMENT(serialnum, bnumber, rnumber)
primary key = (serialnum)
foreign key = (bnumber, rnumber) references ROOM(bnumber,
rnumber)
```

A relational schema

```
EQPTDESC(serialnum, name, colour, weight)
```

does not need to be decomposed.

```
EQPTDESC(serialnum, name, colour, weight)
primary key = (serialnum)
```

The relational schemas EQUIPMENT and EQPTDESC can be synthesised into one schema

```
EQUIPMENT-DESC(serialnum, bnumber, rnumber, name, colour,
weight)
primary key = (serialnum)
foreign key = (bnumber, rnumber) references ROOM(bnumber,
rnumber)
```

The final solution

```
PROJECT(project-title, budget, deadline)
primary key = (project-title)

WORKS-ON(enumber, project-title)
primary key =(enumber, project-title)
foreign key 1 = (enumber) references EMPLOYEE(enumber)
foreign key 2 = (project-title) references
                                   PROJECT(project-title)

BUILDING(bnumber, bname)
primary key (bnumber)
candidate key (bname)

ROOM(bnumber, rnumber, area)
primary key = (bnumber, rnumber)
foreign key = (bnumber) references BUILDING(bnumber)

EMPLOYEE-MANAGER-LOCATION(enumber, first-name, last-name,
mnumber, bnumber, rnumber)
primary key =(enumber)
foreign key 1 = (bnumber, rnumber) references ROOM(bnumber,
rnumber)
foreign key 2 = (mnumber) references
                         EMPLOYEE-MANAGER-LOCATION(enumber)

EQUIPMENT-DESC(serialnum, bnumber, rnumber, name, colour,
weight)
primary key = (serialnum)
foreign key = (bnumber, rnumber) references ROOM(bnumber,
rnumber)
```

End of solution