

Assignment-3: Question-1

Part 1 (4 marks)

For each of the identified FOREIGN KEYS in this schema, **identify any apparent design issues (if any)**. If there are no apparent design issues with a FOREIGN KEY, explicitly state there are no issues with that KEY. (Hint: you may want to sketch out a relational schema diagram for your own reference. This should not be handed in.)

Solution:

- **FOREIGN KEY (manager) REFERENCES Employee (full_name)**
 - This is not correct. Issue is full_name in Employee table is not primary key or unique to be referenced in foreign key.
- **FOREIGN KEY (title) REFERENCES Salary (title)**
 - This is not correct. Issue is title in the salary table is not primary key or unique to be referenced in foreign key.
- **FOREIGN KEY (years_in_role) REFERENCES Salary (years_experience)**
 - This is not correct. Issue is years_experience in the salary table is not primary key or unique to be referenced in foreign key.
- **FOREIGN KEY (employee_sin) REFERENCES Employee (sin)**
 - No issue with the foreign key, 'sin' is a primary key in the 'Employee' table that makes it unique. But having employee_sin attribute as-is in the table OfficeBuilding does not have practical meaning (since with current design only one employee can exist for each building as name_abbrev alone is primary key), it should be either part of primary key along with 'name_abbrev' attribute or should be removed from the table and redesign the table.

Part 2 (6 marks)

Create a new schema (you may do this in the form of CREATE TABLE commands or a relational schema drawn with draw.io or another diagramming tool) which resolves these issues and explain how your solution solves the problems.

Solution:

Below is the new schema along with comments and assumptions for each table. Also I am submitting another text file with just new schema, for testing in SQL.

New Schema: I have highlighted the changes in the schema, like added few new attributes and removed some. Also changed the foreign keys which are also highlighted.

```
CREATE TABLE Salary (  
  salaryID int(9),  
  -- title varchar(12), /*removing it*/  
  -- years_experience int, /*removing it*/  
  salary decimal(15,2),
```

```
PRIMARY KEY (salaryID)
);
```

```
CREATE TABLE Employee (
sin int(9),
title varchar(10),
years_in_role int,
full_name varchar(30) NOT NULL,
-- manager varchar(30), /* removing it*/
manager_sin int(9),
salaryID int(9),
PRIMARY KEY (sin),
office_name varchar(4),
FOREIGN KEY (manager_sin) REFERENCES Employee (sin),
FOREIGN KEY (salaryID) REFERENCES Salary (salaryID),
FOREIGN KEY (office_name) REFERENCES OfficeBuilding (name_abbrev)
-- FOREIGN KEY (manager) REFERENCES Employee (full_name), /*
removing it*/
-- FOREIGN KEY (title) REFERENCES Salary (title), /* removing it*/
-- FOREIGN KEY (years_in_role) REFERENCES Salary (years_experience) /*
removing it*/
);
```

```
CREATE TABLE OfficeBuilding (
name_abbrev varchar(4), -- unique short abbreviation for office
name varchar(20),
-- address
street_number int,
street_name varchar(20),
city varchar(20),
province_state varchar(20),
country varchar(20),
pcode varchar(6),
-- end address
-- employee_sin int(9), /* removing it*/
PRIMARY KEY (name_abbrev)
-- FOREIGN KEY (employee_sin) REFERENCES Employee (sin) /* removing
it*/
);
```

Assumptions/comments for Salary table:

- Created 'salaryID' a new attribute and it is unique for the combination of title and years_experience (business rule).
- Removing the attributes 'title' and 'years-experience' as they are redundant in the salary table, after adding the salaryID.

Assumptions/comments for Employee table:

- Created salaryID a new attribute and it is unique for the combination of title and years_experience (business rule). Also this attribute is foreign key referencing the salaryID in Salary table.
- Since full name will not be unique, it cannot be used as a reference in the foreign key 'manager'. Thus, removing the 'manager' attribute and adding 'manager_sin' attribute.
- Created 'manager_sin' a new attribute and this can be used to query/fetch the manager name. Also this attribute is foreign key referencing the sin in Employee table.

Assumptions/comments for OfficeBuilding table:

- Removing the attribute 'employee_sin', since it is not required in the table and if it is not removed then it should be part of the primary key to make this table practically useful. Having it in the table and not part of primary key means only one employee can be assigned per each office building.
- Accordingly removing the foreign key reference for attribute 'employee_sin'.