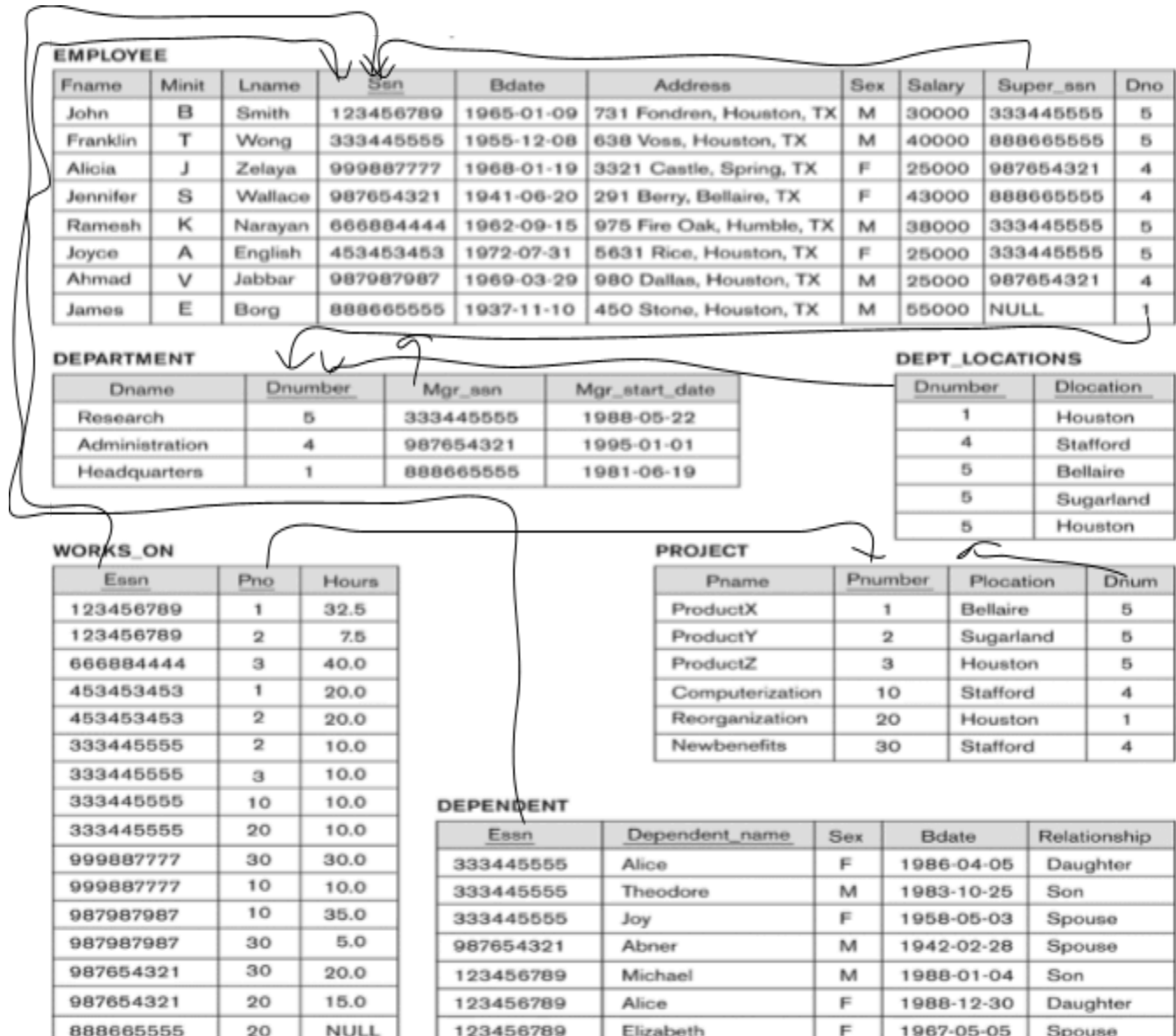


## Practice Sheet

### Chapter 5 (Relational Database Constraints)

#### QUESTION:

Consider the following relational database state, primary keys are underlined, foreign keys are shown with arrow:



Suppose each of the following update operations is applied directly to the above database state. Discuss **which integrity constraint(s) were violated by each operation** and **how**

**you can enforce these constraints to ensure no violation occurs.** If an operation does not violate any constraints, briefly explain why.

- a. Insert <'D1', 'Stafford'> into Department\_Locations table
- b. Insert <6, 'Stafford'> into Department\_Locations table
- c. Insert <null, 'Stafford'> into Department\_Locations table
- d. Update dno of John B Smith to 3 in Employee table.
- e. Delete the employee Franklin T Wong from the Employee table.
- f. Delete the third record from works\_on table
- g. update pnumber of productX to 20 in project table

### **SOLUTION:**

**Your solution explanations should be in your own words.**

- a. Domain constraint, because Dnumber is integer but "D1" is not. Referential Integrity Constraint, because no "D1" in referenced table Department. To prevent violation: Domain constraint -> declare Dnumber as int, Referential Integrity -> declare Dnumber in the Department\_Locations table as foreign key. Then DBMS will disallow this operation.
- b. Referential Integrity Constraint, because no "6" for Dnumber in referenced table Department. To prevent violation: declare Dnumber in the Department\_Locations table as foreign key. Then DBMS will disallow this operation.
- c. Entity Integrity Constraint, Dnumber is part of primary key and so cannot be null. To prevent violation: declare (Dnumber, Dlocation) as primary key when creating the table. Then DBMS will disallow this operation.
- d. Referential Integrity Constraint, because no "3" for Dnumber in referenced table Department. To prevent violation: declare Dno in the Employee table as foreign key. Then DBMS will disallow this operation.
- e. Referential Integrity Constraint, if we notice the SSN of Franklin T Wong, then we can see that this employee is referenced by Department(Mgr\_ssn),

Works\_On(ESSN) and Dependent(ESSN). If we remove this employee from the Employee table, then the values in the referencing tables will not be existing in the referenced table anymore. To prevent violation: when declaring the columns in the referencing table as foreign key also specify what will happen on delete: restrict->deletion from referenced table not allowed, cascade-> the referencing rows will also be deleted or set null-> the value in the referencing column will be set to null.

- f. No violation. This table is not referenced by other tables, so no referential integrity violation. Deletion of a row does not cause other constraint violations in any scenario.
- g. Key constraint, Pnumber is the primary key in the Project table, so duplicate values are not allowed and "20" already exists. To prevent violation, declare pNumber as the primary key during table creation. Also, Referential Integrity constraint, "ProductX" pNumber is "1" which is referenced in the Works\_On table, if the value is updated, then the referencing table's value will not be an existing value from the referenced table any more. To prevent violation: when declaring the columns in the referencing table as foreign key also specify what will happen on update: restrict->update of referenced column in the referenced table not be allowed, cascade-> the referencing rows will also be updated or set null-> the value in the referencing column will be set to null.