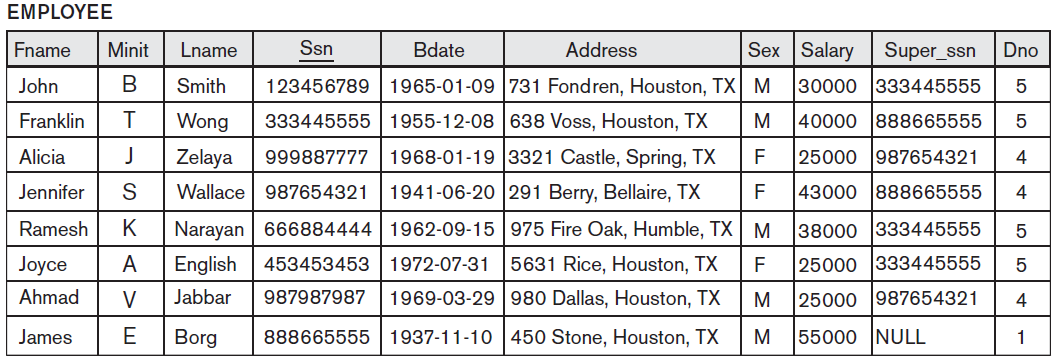
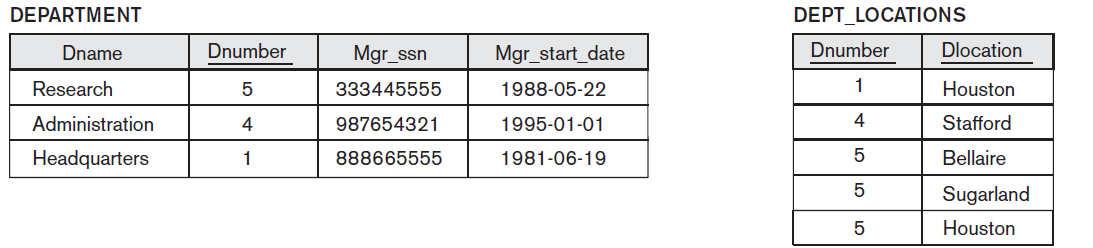
**Chapter 3: THE RELATIONAL DATA MODEL**

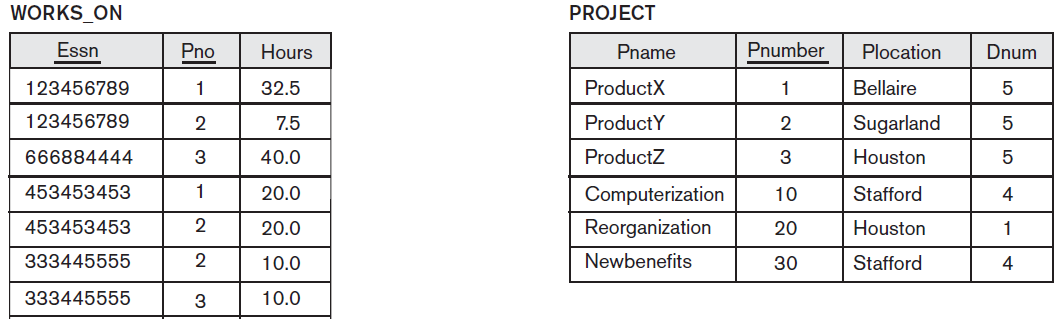
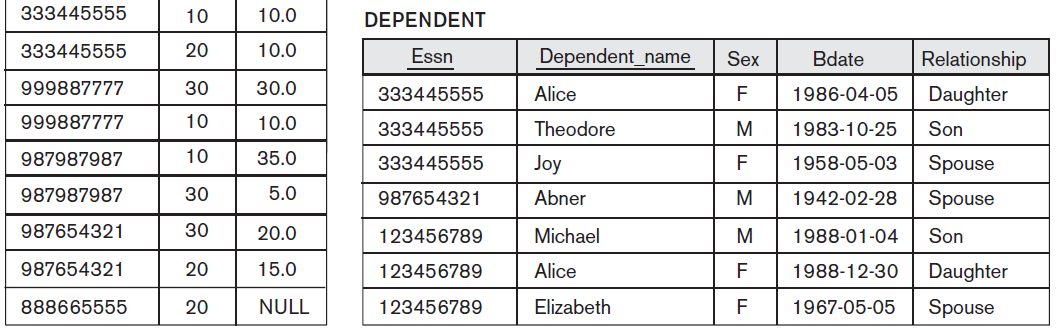
**&**

**RELATIONAL DATABASE CONSTRAINTS**

**Figure:** One possible database state for the COMPANY relational database schema.





**1. Suppose each of the following Update operations is applied directly to the database of above Figure, Discuss *all* integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints:**

1. **Insert <‘Robert’, ‘F’, ‘Scott’, ‘943775543’, ‘1972-06-21’, ‘2365 Newcastle Rd,**

**Bellaire, TX’, M, 58000, ‘888665555’, 1> into EMPLOYEE.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **No** | - | - | - |

1. **Insert <‘ProductA’, 4, ‘Bellaire’, 2> into PROJECT.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **Yes** | - | - | **DNUM:2**  -No tuple in the DEPARTMENT relation with DNUMBER=2 exists.  **Ways of enforcing:**  -Changing the DNUM to either 5, 4, 1  -Inserting a new tuple of DNUM=2 in the DEPARTMENT itself. |

1. **Insert <‘Production’, 4, ‘943775543’, ‘2007-10-01’> into DEPARTMENT.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **Yes** | - | **DNUMBER=4**  -It already exists in the DEPARTMENT tuple.  **Ways of enforcing:**  -Changing the value of DNUMBER which doesn’t exists and also not violate the key constraint. | **MGRSSN='943775543'**  - No tuple exists in the  EMPLOYEE relation with SSN='943775543'.  **Ways of enforcing:**  -Changing the value of MGRSSN to an existing SSN value in  EMPLOYEE.  -Inserting a new EMPLOYEE tuple with SSN='943775543'. |

1. **Insert <‘677678989’, NULL, ‘40.0’> into WORKS\_ON.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **Yes** | **PNO=NULL**  -It cannot be NULL as it is the primary key in WORKS\_ON  **Ways of enforcing:**  -Changing the value of PNO and also the value of PNUMBER should exist in the PROJECT relation. | - | **ESSN='677678989'**  -No tuple exists in the  EMPLOYEE relation with SSN='677678989'.  **Ways of enforcing:**  -Changing the value of ESSN to an existing SSN value in  EMPLOYEE.  -Inserting a new EMPLOYEE tuple with SSN='677678989'. |

1. **Insert <‘453453453’, ‘John’, ‘M’, ‘1990-12-12’, ‘spouse’> into DEPENDENT.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **No** | - | - | - |

1. **Delete the WORKS\_ON tuples with Essn = ‘333445555’.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **No** | - | - | - |

1. **Delete the EMPLOYEE tuple with Ssn = ‘987654321’.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **Yes** | - | - | **Ssn = ‘987654321’**  -Many tuples exist in the WORKS\_ON,  DEPENDENT, DEPARTMENT, and EMPLOYEE relations that reference the tuple being deleted from EMPLOYEE.  **Ways of enforcing:**  -Deleting all tuples in the WORKS\_ON, DEPENDENT, DEPARTMENT, and EMPLOYEE  relations whose values for ESSN, ESSN, MGRSSN, and SUPERSSN. |

1. **Delete the PROJECT tuple with Pname = ‘ProductX’.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **Yes** | - | - | **Pname = ‘ProductX’**  -Affects the other tuple which exists in the WORKS\_ON relates that reference the tuple being deleted from PROJECT.  **Ways of enforcing:**  -Deleting the tuples in the WORKS\_ON relation whose value  for PNO=1 |

1. **Modify the Mgr\_ssn and Mgr\_start\_date of the DEPARTMENT tuple with Dnumber = 5 to ‘123456789’ and ‘2007-10-01’, respectively.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **No** | **-** | **-** | **-** |

1. **Modify the Super\_ssn attribute of the EMPLOYEE tuple with Ssn = ‘999887777’ to ‘943775543’.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **Yes** | **-** | **-** | **SUPERSSN='943775543'**  -No tuple exists in the EMPLOYEE relation with SSN='943775543'.  **Ways of enforcing:**  -Inserting a new EMPLOYEE tuple with SSN='943775543'. |

1. **Modify the Hours attribute of the WORKS\_ON tuple with Essn = ‘999887777’ and Pno = 10 to ‘5.0’.**

|  |  |  |  |
| --- | --- | --- | --- |
| Violation | Entity integrity | Key constraint | Referential integrity |
| **No** | **-** | **-** | **-** |

**2. Consider the relation CLASS (Course#, Univ\_Section#, Instructor\_name, Semester, Building\_code, Room#, Time\_period, Weekdays, Credit\_hours). This represents classes taught in a university, with unique Univ\_section#s. Identify what you think should be various candidate keys, and write in your own words the conditions or assumptions under which each candidate key would be valid.**

Any relation schema may have more than one key. In this case, each of the keys is called a candidate key. It is common to designate one of the candidate keys as the primary keyof the relation. This is the candidate key whose values are used to identify tuples in the relation. All possible candidate keys include the following:

1. **(Univ\_Section#)** - If it is unique across all semesters.
2. **(Semester, BuildingCode, Room#, TimePeriod, Weekdays)** - If more than one course cannot be conducted in the same room during a particular semester.

3. **(InstructorName, Semester) -** If an instructor can only teach one course in each semester.

4. **(Course#, Univ\_Section#, Semeste**r**)** - would be the candidate key, If **(Univ\_Section#)** is not unique across all semesters.