**EXERCISES**

**1. Consider the student data. Suppose that each student has a social security number.**

**Which of the following dependencies are true?**

a) date of birth → student number False

b) student number → date of birth True

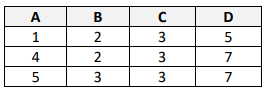
c) student number → shoe size True

d) shoe size → student number False

e) social security number → student number True

f) student number → social security number True

**2. Which of the following functional dependencies do definitely not hold over the table below?**



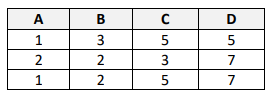
a) A → B

b) B → C

c) {B, C} → A False

d) B → A False

**3. Which of the following functional dependencies do definitely not hold over the table below?**



a) C → B False

b) C → A

c) {C, D} → A

d) A → D False

**4. Determine candidate keys for the following relations:**

**a) R1 (X, Y, Z)**

Z → X

Z → Y

**Candidate Key = Z**

**b) R2 (A, B, C, D)**

C → A

D → B

A → D

**Candidate Key = C**

**c) R3 (A, B, C, D)**

C → D

C → A

B → C

**Candidate Key = B**

**c) R4 (A, B, C)**

B → A

C → B

B → C

C → A**Candidate Key = B & C**

**4. Suppose the following relation:**

**R (A, B, C, D, E)**

**Analyse the given set of dependencies and answer to the questions in the table below.**

**Hint: First, you must determine the primary key for the relation.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Full functional**  **dependency or**  **partial dependency?** | **Transitive or**  **non-transitive**  **dependency?** | **Trivial or non-trivial functional dependency?** |
| **{B, D} → A** | Full | Non-Transitive | Non-Trivial |
| **{B, D} → C** | Full | Transitive | Non-Trivial |
| **{B, D} → E** | Partial | Non-Transitive | Non-Trivial |
| **B → E** | Full | Non-Transitive | Non-Trivial |
| **A → C** | Full | Non-Transitive | Non-Trivial |
| **{B, D} → {B, D}** | Full | Non-Transitive | Trivial |

**5. What is the normal form for each of the relations below? Give arguments!**

**a) CREW\_MEMBER (date, ssn)**

BCNF

**b) PROJECT\_MEMBER (projectno, empno, empRoleInTheProject, gender)**

1NF

**c) COUNTRY (continent, countryname, population)**

BCNF

**d) DEPARTMENT (deptno, deptname, companycode, companyname)**

1NF

**e) TEXTBOOK (ISBN, bookname, chapternumber, chaptername, publisher)**

1NF

**f) EMPLOYEE (empno, surname, firstname, PHONE(phonenumber, phonetype))**

3NF

**g) R (A, B, C, D)**

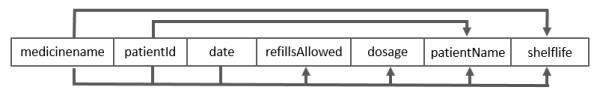
**{A, B} → D**

**{B, C} → D**

**D → A**

3NF

**6. Based on the dependency diagram below, create relations that are in BCNF.**

****

**In your relation schemas,**

* **Underline primary keys and include all foreign key definitions: FK (a) REFERENCES r2 (a)**
* **Show functional dependencies in your relations (use this notation: A → B)**

Patient(patientId, patientName)

Medicine(medicineName, shelfLine)

Prescription(medicineName, patientId, date, refillsAllowed, dosage)

FK (patientId) REFERENCES Patient (patientId)

FK (medicineName) REFERENCES Medicine (medicineName)

{ medicineName, patientId, date }→{ refillsAllowed, dosage }

medicineName→ shelfLine

patientId→ patientName

**7.**

1. PLAYER (playerno, surname, firstname, teamnumber, teamname)  
   The normal form of the relation is 2NF  
   Player(playerno, surname, firstname, teamno)  
   FK (teamno) REFERENCES Team(teamno)  
   Team(teamno, teamname)  
   playerno →{ surname, firstname, teamno }  
   teamno→ teamname
2. BOOKING (ISBN, bookname, patronid, surname, firstname, bookingdate, bookingtime)  
   The normal form of the relation is 1NF  
   Booking(ISBN, patronid, bookingdate, bookingtime)  
   FK (ISBN) REFERENCES Book(ISBN)  
   FK (patronid) REFERENCES Patron(patronid)  
   Book(ISBN, bookname)  
   Patron(patronid, surname, firstname)  
   ISBN →Bookname  
   patronid→{ surname, firstname }
3. ORDER (orderno, productno, productname, quantity, clientno, deliveryaddress, clientaddress)  
   The normal form of the relation is 1NF  
   Order(orderno, productno, quantity, clientno, deliveryaddress)  
   FK (clientno)REFERENCES Client(clientno)  
   FK (productno)REFERENCES Product(productno)  
   Client(clientno, clientaddress))  
   Product(productno, productname)  
   { orderno, productno }→{ quantity, clientno, deliveryaddress }  
   clientno → clientaddress  
   productno→productname

# **8.**

1. a is primary key
2. There is a transitive dependency in the relation : a→b→d
3. XYZ(a,b,c,e)  
   OPQ(b,d)

**9.**

1. {a, d} is primary key
2. There is a partial dependency in the relation : a→c
3. XYZ(a,b,d)  
   OPQ(a,c)