23K-0004
Syed Huzaifa Ali

Database Systems

Assignment # 02

Q1.

A.

Film:
FilmID (PK)
Title
Budget
ReleaseDate
DirectorName

Actor:
ActorID (PK)
ActorName
Phone

Company:
CompanyID (PK)
CompanyName

Casting:
FilmID (FK)
ActorID (FK)
CharacterName

Film_Funding:
FilmID (FK)
CompanyID (FK)
Amount

B.
Film – Actor → M:N (Many to Many)
Film - Company → M:N
Film - Director → 1:1 (one to one)

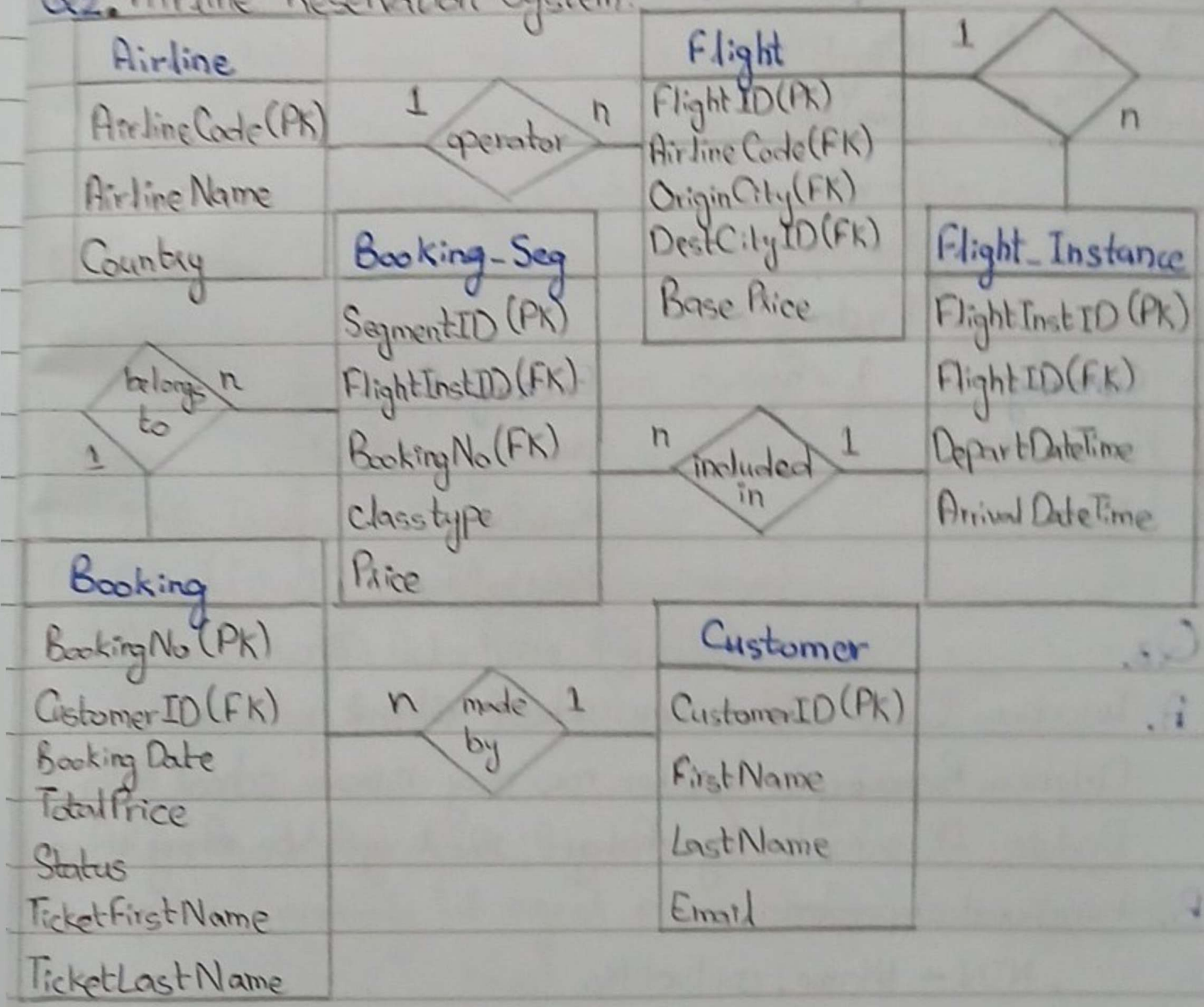
C. Casting (Name)
FilmID (FK - Film) Part of composite PK.
ActorID (FK - Actor) Part of composite PK.
CharacterName - Not NULL

D. In this case, we don't need full details about directors, only their names for each film, and the problem explicitly states director is not tracked separately; director

23K-0004

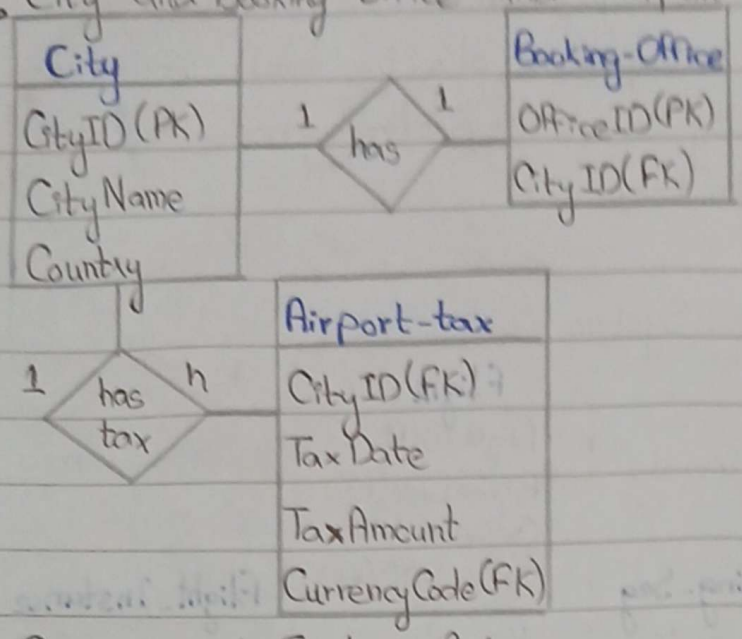
data only stored as text per film. It is not reused in other records, so creating a separate director table would add unnecessary complexity so we keep directorname as normal attribute inside Film table.

Q2. Airline Reservation System:

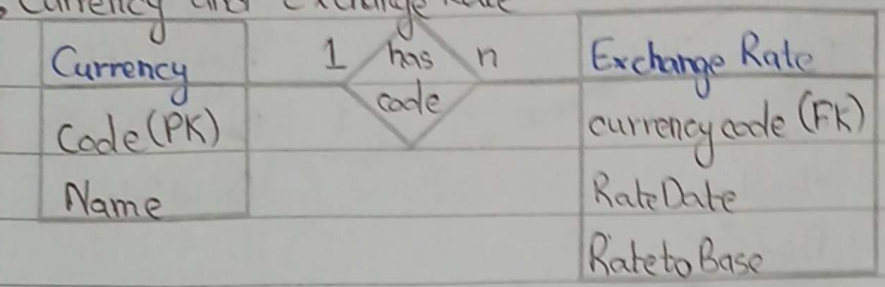


23K-0004

- City and Booking Office and Airport tax



- Currency and Exchange Rate



Q3.

A. Insertion: Can't add a new school without a teacher.
Deletion: Removing a teacher row may remove school info.
Update: If school city changes, must update many rows.

3. Functional dependencies:
   - NIN → tName, contactNo
   - SchoolID → SchoolName, Schoolcity
   - (NIN, schoolID) → hours.

Normalization Steps:

① 1NF → already atomic

② 2NF → remove partial dependencies.
   - Teacher (NIN, tName, contactNo)
   - School (SchoolID, SchoolName, Schoolcity).

23K-0004

- Teach-Assignment (NIN, schoolID, hours)

③ 3NF → No transitive dependencies.

Q4.

A. Primary keys : (FilmID, ActionID).

B. Violates (2NF) in 1NF, all attributes have atomic values, but in this table, some non-key attributes depend only on part of the composite primary key, not the whole key. Thats a partial dependency which breaks 2NF rules.

C. Examples: If the tables primary key is (FlightNo, Date) then an attribute like AirlineName or OriginCity depends on FlightNo, not on Date.

C. 2NF. Tables:

Film (FilmID, FilmTitle, DirectorName)

Actor (ActorID, ActorName, AgentID)

Film_casting (FilmID, ActorID)

D. 3NF:

Add Agent (AgentID, AgentPhone).

E. Update anomaly (if agent phone changes, must edit all rows).

23K-0001

Q5.

1. Assumptions:

Student has unique ID.

Course has unique couse ID.

Student can enroll in many courses and a course can have many students.

Functional Dependencies:

Student ID → First Name, Last Name, Address

Course ID → Course Name, credits, Instructor

(Student ID, Course ID) → Enrollment Date

2. Already in 1NF.

1NF → 2NF:

Student (Student ID, First Name, Last Name, Address).

Course (Course ID, Course Name, Credits, Instructor Name).

Enrollment (Student ID, Course ID, Enrollment Date).

2NF → 3NF: Check for transitive dependencies.

All tables resulting from the 2NF process are already in 3NF.

Student (Student ID, First Name, Last Name, Address)

Course (Course ID, Course Name, credits, Instructor Name)

Enrollment (Student ID, Course ID, Enrollment Date).

3. Primary Keys:

Student → Student ID

Course → Course ID

No Alternate Keys.

Foreign Keys:

Student ID (FK to Students)

Course ID (FK to Course)