Ibrahim Johar Farooqi
23K-0074
BAI-5A
DBS Assignment 2

day / date:

Q.1) Part A

Entities	Attributes
Films	FilmID, Title, Budget, ReleaseDate, DirectorName
Actors	ActorID, Name, PhoneNumber
Production_Company	CompanyID, Name
Casting	FilmID, ActorID, CharacterName

↳ Part B

↳ Films ~~casts~~ casts → Actors : M:N (Many to Many)
↳ a film casts many actors since a film has many actors and an actor ~~casts~~ can work in many films.

↳ Films funded by ↔ ProductionCompany : M:N (many to many)
↳ a film can be funded by many companies, and a company can fund multiple films

↳ Film directed by ↔ Director : 1:1
↳ every film has exactly one director

↳ Part C

↳ Associative Entity : Casting
↳ Attributes : FilmID, ActorID, CharacterName
↳ Primary Key : Will be composite, comprising of FilmID & ActorID
↳ Foreign Key : ActorID → references ~~Film~~ Actor
FilmsID → references Film

↳ Part D

↳ Rule 4 states ~~that a film can~~ that each film has only one director. There are no additional attributes for the director to be stored, and there is no need to track directors independently as they are not involved in any relationships like funding, acting, multiple films. Creating a separate Director table would only entail further unnecessary complexity, and will provide no new relational benefit.

day / date:

Q2) Attached separately

Q.3)

NIN	contractNo	hours	tName	schoolID	schoolName	schoolCity
2103	T9001	10	Brown A	S301	Park School	Manchester
2198	T9001	18	Khan R	S301	Park School	Manchester
2205	T9001	20	Lewis H	S114	Green School	Birmingham
2103	T9001	12	Brown A	S114	Green School	Birmingham

a) Insertion Anomaly

↳ We cannot insert a new school into the table without adding dummy values for NIN, hours, etc.

↳ like even if we want to insert a new teacher who is not assigned a school, that will not be possible.

↳ 2206 T9001 14 'John Mark' NULL NULL NULL

Deletion Anomaly

↳ If we delete the only row containing information about a certain school, like Park School, like if we remove the only row of information for Park School → (2198, T9001, 18, Khan R, S301, Park School, Manchester), we will lose Park School's name and city information from the database.

Updation Anomaly

↳ If we were to update the school city for (S301, Park School) from Manchester to London, we would have to change/update every instance of Park School in the database, failure to do so would lead to data inconsistency.

$A \rightarrow B$

↳ "B depends on A"

↳ "A uniquely/functionally determines B"

day / date:

b) ↳ step1: determining functional dependencies

1. ↳ NIN → tName
   (a teacher's NIN uniquely identifies their name)
2. ↳ schoolID → schoolName, schoolCity
   (schoolCity & schoolName depend on schoolID)
3. ↳ (NIN, contractNo, schoolID) → hours

* we are assuming contractNo does not determine any other attribute on its own, so we will not create a seperate table for it.

↳ primary key of original table → composite key of (NIN, schoolID)

↳ step2: convert to 2NF (if in 1NF already)

↳ removing partial dependencies; when a non-key depends on only part of a composite key

↳ seperate school & teacher information

↳ Teacher - Table (After 2NF):

NIN (PK)	tName
2103	Brown A
2198	Khan R
2205	Lewis H

↳ school - Table (After 2NF):

SchoolID (PK)	schoolName	schoolCity
S301	Park School	Manchester
S114	Green School	Birmingham

↳ TeacherAssignment - Table

NIN (FK)	schoolID (FK)	hours	contractNo
2103	S301	10	T9001
2198	S301	18	T9001
2205	S114	20	T9001
2103	S114	12	T9001

composite key: (NIN, schoolID, contractNO)

day / date:

↳ step3: convert to 3NF

↳ removing transitive dependencies (non-key attributes depending on other non-key attr.(s))
↳ keeping attributes at all times dependent on the key

↳ Teacher (NIN, tName)
↳ PK: NIN
NIN → tName

↳ School (schoolID, schoolName, schoolCity)
↳ PK: schoolID
schoolID → schoolName, schoolCity

↳ TeacherAssignment (NIN, ContractNo, schoolID, hours)
↳ PK: NIN, schoolID, contractNo
↳ FK: NIN, schoolID
(NIN, schoolID, contractNo) → hours

↳ already in 3NF.

(Q.4) Functional Dependencies of FILM_CASTING:
↳ FilmID → FilmTitle, DirectorName
↳ ActorID → ActorName, AgentID
↳ AgentID → AgentPhone
↳ FilmID, ActorID → all other attributes (no other single attribute determines the whole record)

a) Primary Key: composite Primary key consisting of (FilmID & ActorID)

b) The table violates the 2NF, as the non-key attributes like DirectorName depend on 'FilmID' and not on the entire composite primary key, causing partial dependency, leading to violation of 2NF.

c) 2NF applied
① ↳ Film [FilmID, FilmTitle, DirectorName]
↳ Dependency removed:- (FilmID → FilmTitle, DirectorName)
② ↳ Actor [ActorID, ActorName, AgentID]
↳ Dependency removed:- (AgentID → ActorName, AgentID)
③ ↳ FilmCasting [FilmID, ActorID]
↳ no partial dependency.

day / date:

d) Transitive dependency exists in 'Actor' table, since the non-key attribute 'AgentPhone' depends on another non-key attribute 'AgentID', which leads to dependency on primary key ActorID — that will lead to indirect determination of AgentPhone through AgentID.

↳ after 3NF :

↳ Agent [AgentID(PK), AgentPhone]

↳ Film [FilmID(PK), FilmTitle, DirectorName]

↳ Actor [ActorID(PK), AgentID(FK), ActorName]

↳ FilmCasting [FilmID(FK), ActorID(FK)]

↳ remaining transitive dependencies removed

e) Update Anomaly, as the attribute 'AgentPhone' is stored repeatedly for each film the actor is starring in (included in). Consequently, if the Agent's phone number changes, we will have to update it in multiple rows, causing a potential for inconsistency in data.

Q.5) 1) Functional Dependencies

↳ StudentID ⟶ FirstName, LastName, Address

↳ CourseID ⟶ CourseName, Credits, InstructorName

↳ (StudentID, CourseID) ⟶ EnrollmentDate

Assumptions

↳ StudentID & CourseID are unique

↳ each course is taught by one instructor

↳ one student can enroll in multiple courses

↳ enrollment depends on studentID & courseID.

2) ↳step 1: table is in 1NF as all values are atomic

studentID, FirstName, LastName, Address, CourseID, CourseName, Credits, InstructorName, EnrollmentDate

↳step 2: partial dependencies to be removed

day / date:

↳ Student (StudentID, FirstName, LastName, Address)
↳ Course (CourseID, CourseName, Credits, InstructorName)
↳ Enrollment (CourseID, StudentID, EnrollmentDate)

↳ step 3: transitive dependencies to be removed (3NF)
↳ no transitive dependencies to be removed

3)

Tables	Primary Key	Foreign Key	Alternate Key
Student	StudentID	—	—
Courses	CourseID	—	—
Enrollment	StudentID, CourseID	StudentID → Student	—
		CourseID → Course	