COMP 3005
Assignment #5
Due: December 4

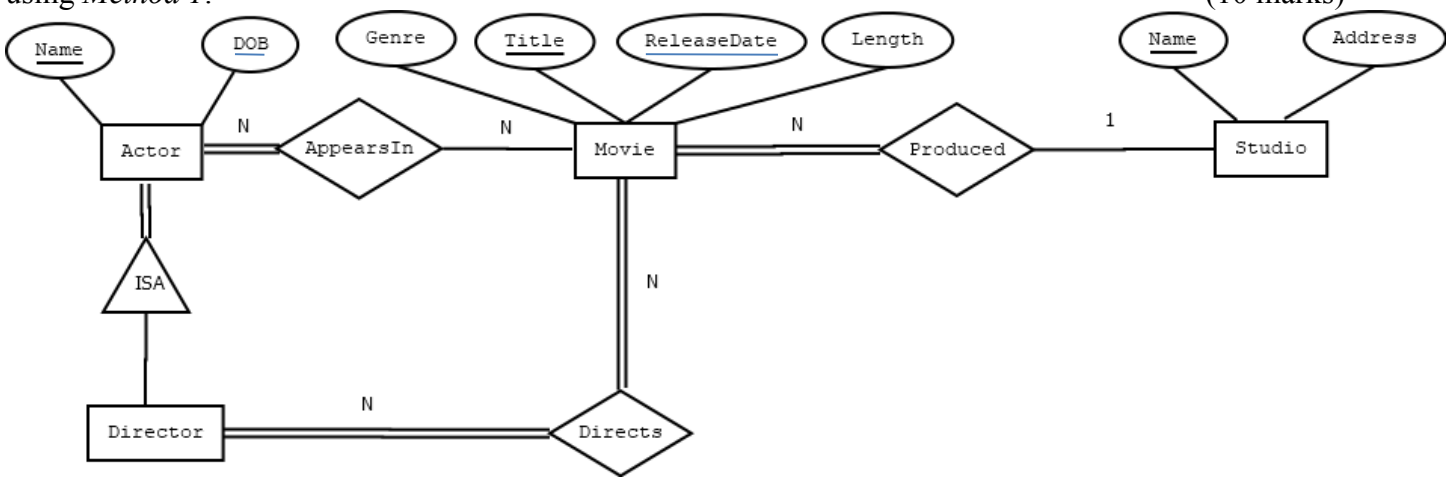
Instruction

1. This is an individual assignment. Copying is not allowed.
2. Do not use SQL to create any table for this assignment.
3. You need to use Oracle VM to do Part 4 of this assignment and submit the final version of the program together with the screenshots that contain the file name you type to execute the program and the execution results the system display. If there is no screenshot, you will not get any mark for the question.
4. Submit your assignment as a single word/pdf document on culearn by the due date.

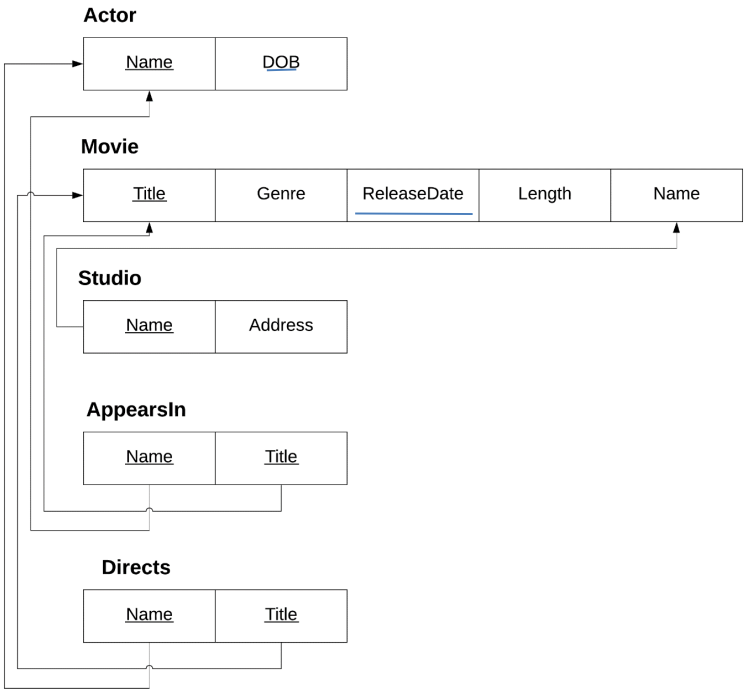
Part 1 (25 marks)

Movie enterprises include movies, actors, directors, and studios. Each movie is identified by title and year of release, has a length in minutes, and is classified under one or more genres such as horror, action, drama. It is produced by one studio and has one or more directors and zero or more actor. Actors are identified by name and data of birth and appear in one or more movies. Directors are also identified by name and date of birth and direct one or more movies. It is possible for a director to act in a movie, including one that he or she may also direct. Studios have an address and are identified by name. Each of them produces one or more movies.

1. Design an ER/EER diagram that describes this database application with all relevant constraints represented using *Method 1*. (10 marks)



2. Convert the above design using *Method 3*. (5 marks)
   For double line N, change to thick line and delete N; for single line N, change to thin line and delete N; for double line 1, change to thick arrow in the diagram and delete 1, for single line 1, change to a thin line arrow and delete 1.
3. Map the EER diagram into the relational schema that consists of relation names, their attributes with primary keys underscored and foreign keys pointing to the corresponding attributes properly (8 marks)

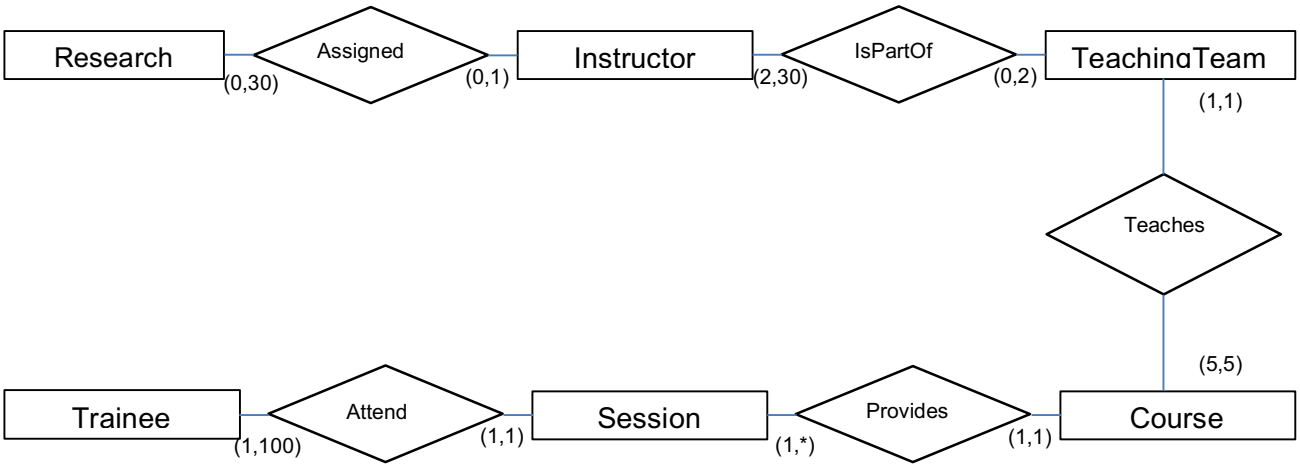


4. Specify which constraints are not supported by SQL. (2 marks)
has one or more directors and zero or more actor

Part 2 (20 marks) Database systems a practical approach page 346

A training company has 30 instructors and handle up to 100 trainees per training session. The company offers five advanced IT courses, each of which is taught by a teaching team of two or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee takes one such course per training session.

1. Design an ER/EER diagram that describes this database application with all relevant constraints represented using *Method 2*. You need to specify your assumption and introduce attributes accordingly. (10 marks)
Deduct 2 for missing an entity type or adding an extra entity type, deduct one for wrong cardinality constraint on one side of the relationship.
Assume research has topics, instructor has id (unique) and name, teaching team has name (unique), course has name (unique) , session has name (unique), trainee has id (unique) and name. They are not yet in the ER diagram below



Note company should not be an entity, just image how many instances it can have.

2. Map the ER/EER diagram into the relational schema that consists of relation names, their attributes with

primary keys underscored and foreign keys pointing to the corresponding attributes properly (8 marks)
Missing

3. Specify which constraints are not supported by SQL. (2 marks)
Each instructor is assigned to a maximum of two teaching teams

Part 3 (35 marks)

1. Consider the following relation. Which of the following functional dependencies may hold in this relation? If the dependency cannot hold, explain why by specifying the tuples that cause the violation. (5 marks)

A	B	C
a1	b1	c1
a2	b2	c1
a4	b1	c2
a3	b4	c3
a1	b1	c4
a3	b4	c5

(a) A → B hold

(b) B → C not hold as b1 has c1,c2,c4

(c) C → B not hold as c1 has b1,b2

(d) B → A not hold b1 has a1,a4

(e) C → A not hold as c1 has a1,a2

2. Given a relation schema for schedules of courses, sections, and instructors at a university and the functional dependencies as follows:

Schedule

C#	S#	I#	Dept	Credit	Level	Term	Year	LOC	No of Students

C# → Dept, Credit, Level

C#, S#, Term, Year → I#, LOC, No of Students

LOC, Term, Year → C#, S#, I#

a. Use inference rules to find out sets of attributes forming candidate keys of **Schedule**. (6 marks)

(1) C# → C#, Dept, Credit, Level with R1

(2) C#,S#,Term,Year → I#, LOC, No of Students and C# → C#, Dept, Credit, Level, we get:
C#,S#,Term, Year → C#, S#, Term, Year, I#, LOC, No of Students, Dept, Credit, Level (all attributes)

(3) LOC, Term, Year → C#, S#, I#, then LOC, Term ,Year→LOC, Term, Year, C#, S#, I#. With C#→Dept, Credit, Level, we have LOC, Term, Year → C#, S#, I#, then LOC, Term ,Year→LOC, Term, Year, Dept, Credit, Level. We miss one No. of Students.. Finally, since C#,S#,Term,Year are now all in the list, and C#,S#, Term, Year → No. of Students, we get:
LOC,Term, Year→ LOC, Term, Year, I#, C#, S#, Dept, Credit, Level, No of Students (all attributes)
Hence, both K1 = {C#, S#, Term, Year} and K2 = {LOC, D, Term, Year} are keys of the relation.

b. Normalize this relation into BCNF. (4 marks)

By applying the general definition of 2NF, we find that the functional dependency {C} →{DEPT, CREDIT, LEVEL} is a partial dependency for K1 (since C is included in K1). Hence, Schdule is normalized into schedule1 and Schedule2 as follows:
Schedule1 = {C#,DEPT, CREDIT, LEVEL}
Schedule2 = {LOC, D, Term, Year, I#,C#,S#, No of Students} with candidate keys K1 and K2
Since neither Schedule1 nor Schedule2 have transitive dependencies on either of the candidate keys, Schedule1 and Schedule2 are in 3NF also. They also both satisfy the definition of BCNF.

3. Given a database consisting of one first normal form relation R with attributes A, B, C, D, E, and the functional and multi-valued dependencies as well as the functional and the multi-valued dependency diagram as follows:

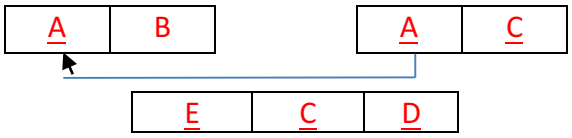
R	A	B	C	D	E

Functional Dependencies {BC → A, A → B}
Multivalued Dependencies {E →→ C, E→→ D}

a. What is the key for R? (2 marks)
ACDE

b. What is the highest normal form this relation is in? (2 marks)
3NF

c. Normalize the relation into BCNF by giving its relation names, their attributes with primary keys underscored and foreign keys pointing to the corresponding attributes properly. (3 marks)



d. Normalize the result in c into the fourth normal form by giving the relation names, their attributes with primary keys underscored and foreign keys pointing to the corresponding attributes properly. (3 marks)

Note that this assignment is different from previous one and thus the solution is different

4. Given the following inference rules:
(R0) X → XX
(R1) if X ⊇ Y, then X → Y
(R2) X → Y ⊨XZ → YZ
(R3) X → Y, Y → Z ⊨ X → Z
Prove or disprove the following inference rules for functional/multi-valued dependencies.

a. X → Y |= Y → X (2 marks)
x1,y1
x2,y1

b. X → Y and Z ⊆ Y and W ⊆ Z |= X → W (2 marks)
Z ⊆ Y then Y → Z by R1
W ⊆ Z then Z → W by R1
X → Y, Y → Z, Z → W then X → W by R3

c. X → Y, XY → Z ⊨ X → Z (2 marks)
X → Y then XX → XY by R2
X → XX, XX → XY, XY → Z then X → Z by R0 and R3

d. X → Y, Z → W ⊨ XZ → YW (2 marks)
X → Y then XZ → YZ by R2
Z → W then YZ → YW

XZ → YZ, YZ → YW then XZ → YW by R3

e. XY → Z, X → YV, Z → W ⊨ X → W (2 marks)

YV → Y by R1,

X → YV, YV → Y then X → Y by R3

X → Y then XX → XY by R2

X → XX by R0, XX → XY, XY → Z, Z →W then X → W

Part 4 (30 Marks)

This part is based on the Person-Hobby database used for previous assignment with three tables shown below.

1. Write an Embedded SQL program that uses dynamic SQL method 1 to create this database. You can hard code all create table statements in the program using the execute immediate statement. (5 marks)
   (2 for the program and 3 for the screenshots).
2. Write an Embedded SQL program that uses dynamic SQL method 2 to insert all tuples into the database. It should first display all table names and prompt the user to choose a table and then prompt the user with the attribute names and accept valid values for each tuple. (10 marks)
   (4 for the program and 6 for the screenshots)
3. Write an Embedded SQL program that uses dynamic SQL method 3 to list all person rows, in person number order. Each person row should be immediately followed in the listing by all hobby rows for hobbies that the person has, in hobby number order. Persons without any hobby should still be listed. (15 marks)
   (5 for the program and 6 for the screenshots)

Person

P#	Name	Age
P1	Smith	20
P2	Jones	30
P3	Blake	25
P4	*Lastname*	20
P5	Adams	30

Hobby

H#	Name
H1	Bowling
H2	Chess
H3	Dancing
H4	Hiking
H5	Skating
H6	Ski

Play

P#	H#	Times
P1	H1	3
P1	H2	2
P1	H3	4
P1	H4	2
P1	H5	1
P1	H6	1
P2	H1	3
P2	H2	4
P2	H3	5
P2	H4	2
P3	H2	2
P3	H3	3
P4	H2	3
P4	H3	4