CSE1500 - WEB AND DATABASE TECHNOLOGY

DB LECTURE 3 - EXERCISES

RELATIONAL SCHEMA DESIGN SCENARIOS

Christos Koutras

LIBRARY

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher, for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names. Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has. Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store.

LIBRARY – IDENTIFYING ENTITIES

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher, for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names. Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has. Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store.

LIBRARY – IDENTIFYING ENTITIES

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher, for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names. Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has. Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store.

LIBRARY – STATE AFTER ENTITY IDENTIFICATION

BOOK

PUBLISHER

LIBRARY BRANCH

BORROWER

LIBRARY – IDENTIFYING ATTRIBUTES

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher, for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names. Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has. Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store.

LIBRARY — IDENTIFYING ATTRIBUTES

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher, for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names. Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has. Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store.

LIBRARY – STATE AFTER ATTRIBUTE IDENTIFICATION

BOOK (ID, Title**)**

PUBLISHER (Name, Address, Phone**)**

LIBRARY BRANCH (ID, Name, Address**)**

BORROWER (Card Number, Name, Phone**)**

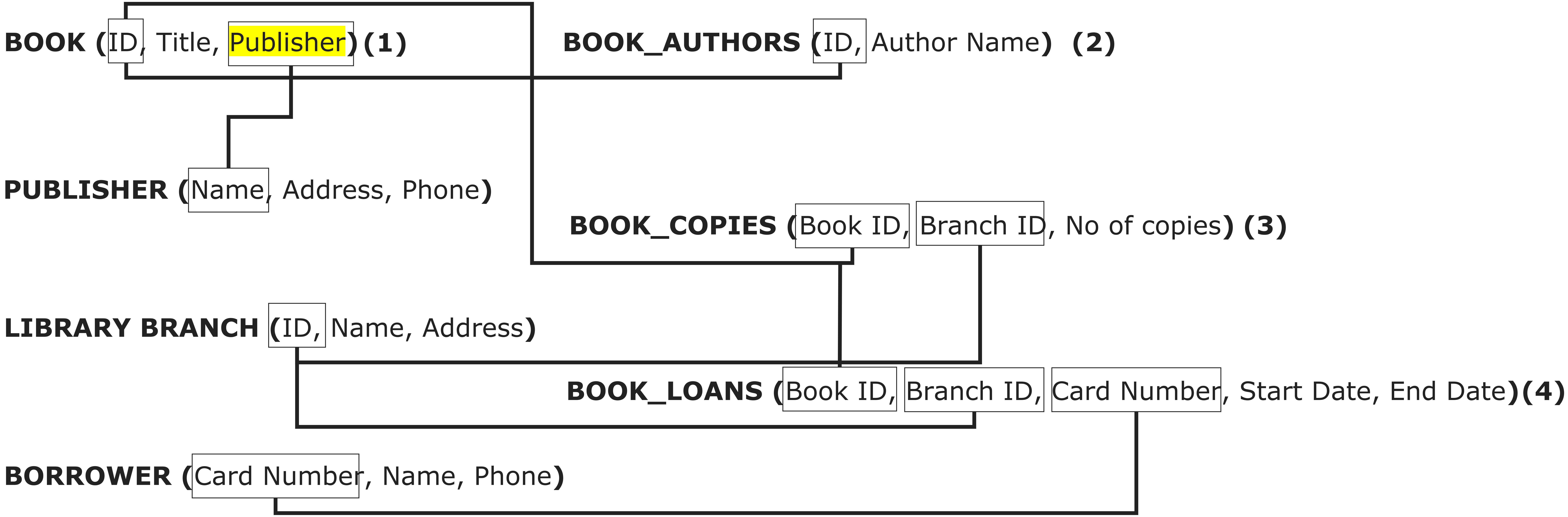
LIBRARY – IDENTIFYING RELATIONSHIPS

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher, for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names. Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has. Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store.

LIBRARY – IDENTIFYING RELATIONSHIPS

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher (1), for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names (2). Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has (3). Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store (4).

LIBRARY – STATE AFTER RELATIONSHIP IDENTIFICATION



BOOK (ID, Title, Publisher**) (1)**

BOOK_AUTHORS (ID, Author Name**) (2)**

PUBLISHER (Name, Address, Phone**)**

BOOK_COPIES (Book ID, Branch ID, No of copies**) (3)**

LIBRARY BRANCH (ID, Name, Address**)**

BOOK_LOANS (Book ID, Branch ID, Card Number, Start Date, End Date**)(4)**

BORROWER (Card Number, Name, Phone**)**

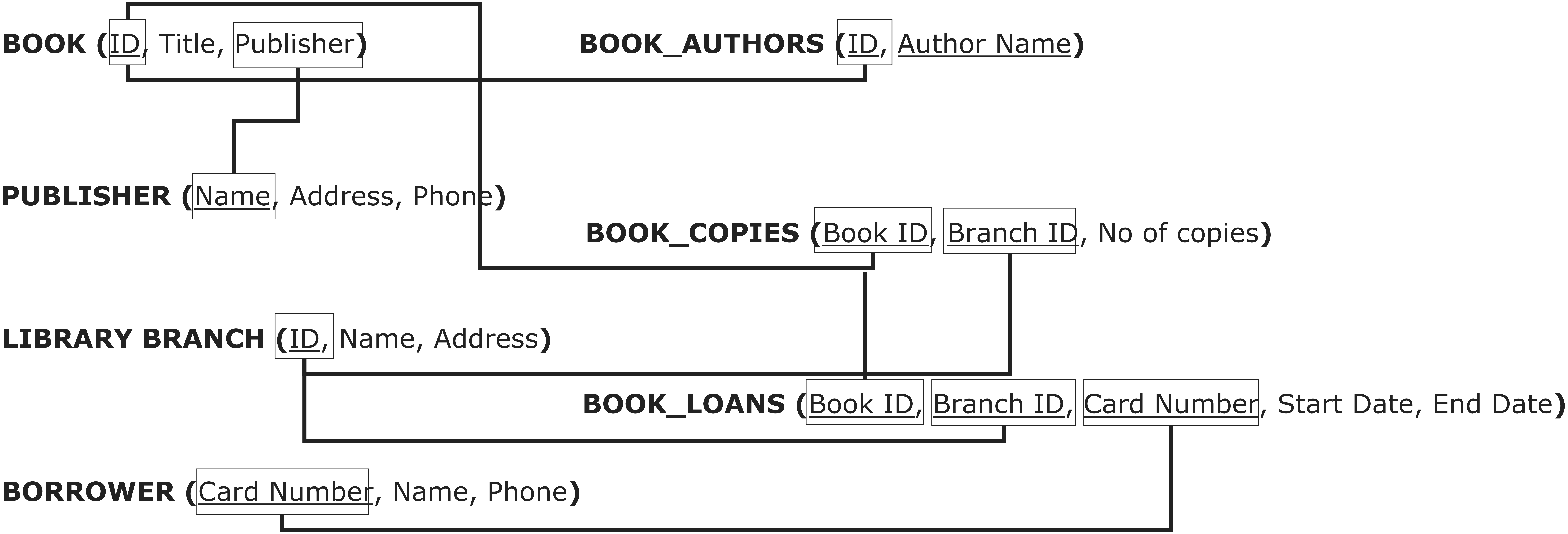
LIBRARY – IDENTIFYING KEYS

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher, for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names. Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has. Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store.

LIBRARY – IDENTIFYING KEYS

We want to design a system that handles book loans in a library. Specifically, for each book we keep information about its ID and title. A book has exactly one publisher, for whom we store their name (unique per publisher), address and phone. In addition, a book may be associated with one or more author names. Books can be found in the library's branches, for which we store an ID, name and address. For each such branch, the system should keep the number of copies of each book it has. Moreover, we want to keep track of the book loans: a borrower, for whom we store their name, phone and identifying card number, can loan a book in a branch of the library for a specific start and end date which we store.

LIBRARY – STATE AFTER KEY IDENTIFICATION



BOOK (ID, Title, Publisher**)**

BOOK_AUTHORS (ID, Author Name**)**

PUBLISHER (Name, Address, Phone**)**

BOOK_COPIES (Book ID, Branch ID, No of copies**)**

LIBRARY BRANCH (ID, Name, Address**)**

BOOK_LOANS (Book ID, Branch ID, Card Number, Start Date, End Date**)**

BORROWER (Card Number, Name, Phone**)**

SOCCER TOURNAMENT

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the date their starting and ending date of the contract. For each match, we store the referee, with first name, surname (different for each referee), city and region of birth; each match has exactly one referee. A match takes place in a stadium, where for each stadium we keep track of its identifying name, capacity and coordinates.

SOCCER TOURNAMENT – IDENTIFYING ENTITIES

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the date their starting and ending date of the contract. For each match, we store the referee, with first name, surname (different for each referee), city and region of birth; each match has exactly one referee. A match takes place in a stadium, where for each stadium we keep track of its identifying name, capacity and coordinates.

SOCCER TOURNAMENT – IDENTIFYING ENTITIES

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the date their starting and ending date of the contract. For each match, we store the referee, with first name, surname (different for each referee), and region of birth; each match has exactly one referee. A match takes place in a stadium, where for each stadium we keep track of its identifying name, capacity and coordinates. Finally, we keep information for the seats of each stadium, namely seat number (unique per stadium), type and booking cost.

SOCCER TOURNAMENT – STATE AFTER ENTITY IDENTIFICATION

MATCH

TEAM

REFEREE

STADIUM

SEAT

PLAYER

SOCCER TOURNAMENT – IDENTIFYING ATTRIBUTES

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the date their starting and ending date of the contract. For each match, we store the referee, with first name, surname (different for each referee), and region of birth; each match has exactly one referee. A match takes place in a stadium, where for each stadium we keep track of its identifying name, capacity and coordinates. Finally, we keep information for the seats of each stadium, namely seat number (unique per stadium), type and booking cost.

SOCCER TOURNAMENT – IDENTIFYING ATTRIBUTES

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the starting and ending date of the contract. For each match, we store the referee, with first name, surname (different for each referee), and region of birth; each match has exactly one referee. A match takes place in a stadium, where for each stadium we keep track of its identifying name, capacity and coordinates. Finally, we keep information for the seats of each stadium, namely seat number (unique per stadium), type and booking cost.

SOCCER TOURNAMENT – STATE AFTER ATTRIBUTE IDENTIFICATION

MATCH (ID, Date, Start Time, Score**)**

TEAM (Name, City, Trainer**)**

REFEREE (First Name, Surname, Birth Region**)**

STADIUM (Name, Capacity, Coordinates**)**

SEAT (Number, Type, Cost**)**

PLAYER (First Name, Surname, Birth date, Position**)**

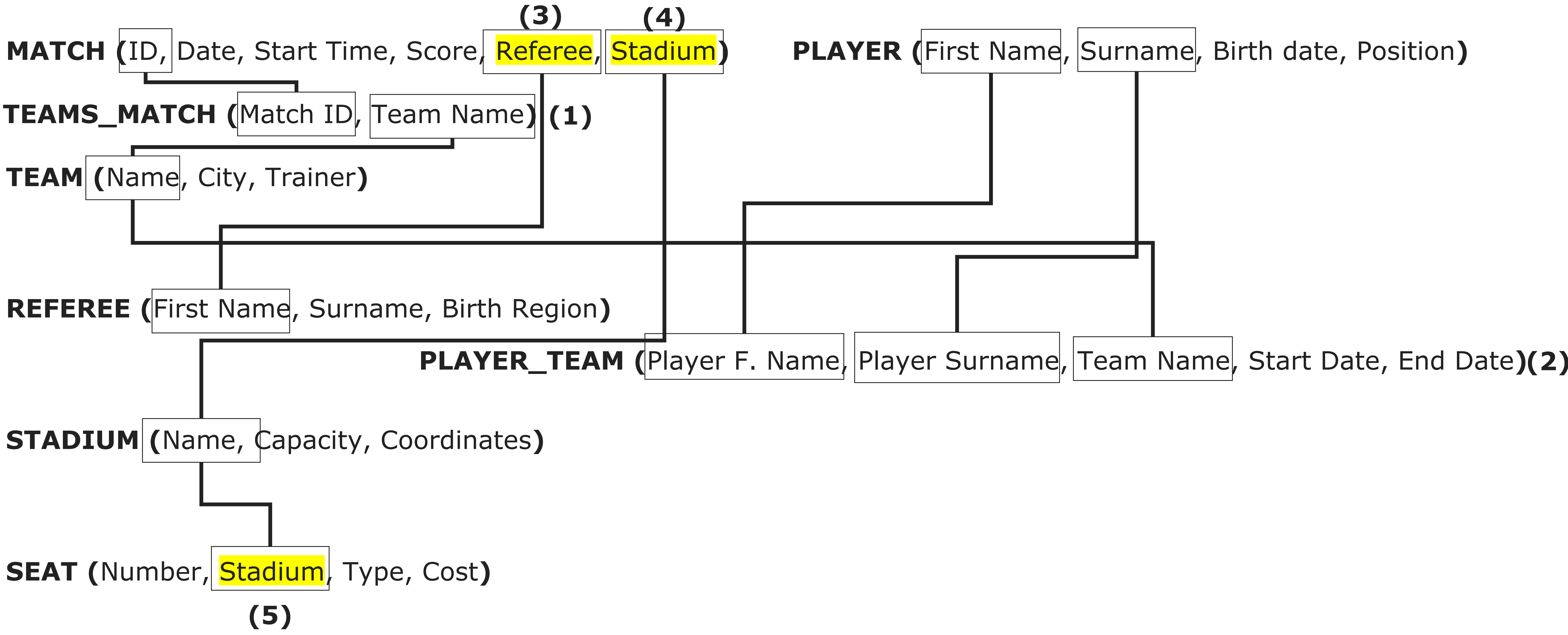
SOCCER TOURNAMENT – IDENTIFYING RELATIONSHIPS

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the starting and ending date of the contract. For each match, we store the referee, with first name, surname (different for each referee), and region of birth; each match has exactly one referee. A match takes place in a stadium, where for each stadium we keep track of its identifying name, capacity and coordinates. Finally, we keep information for the seats of each stadium, namely seat number (unique per stadium), type and booking cost.

SOCCER TOURNAMENT – IDENTIFYING RELATIONSHIPS

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating (1) teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the starting and ending date of the contract (2). For each match, we store the referee, with first name, surname (different for each referee), and region of birth; each match has exactly one referee (3). A match takes place in a stadium (4), where for each stadium we keep track of its identifying name, capacity and coordinates. Finally, we keep information for the seats of each stadium (5), namely seat number (unique per stadium), type and booking cost.

SOCCER TOURNAMENT – STATE AFTER RELATIONSHIP IDENTIFICATION



MATCH (ID, Date, Start Time, Score, Referee, Stadium**)** (3) (4)

PLAYER (First Name, Surname, Birth date, Position**)**

TEAMS_MATCH (Match ID, Team Name**)** **(1)**

TEAM (Name, City, Trainer**)**

REFEREE (First Name, Surname, Birth Region**)**

PLAYER_TEAM (Player F. Name, Player Surname, Team Name, Start Date, End Date**)(2)**

STADIUM (Name, Capacity, Coordinates**)**

SEAT (Number, Stadium, Type, Cost**)**

(5)

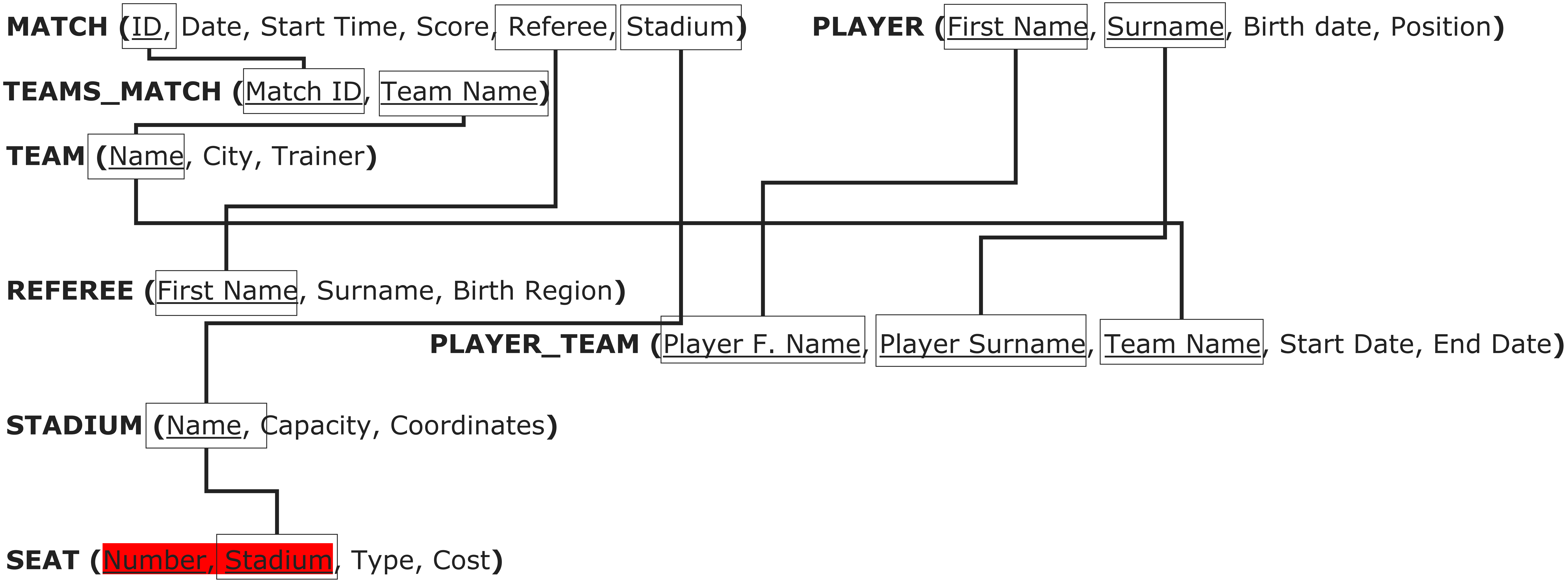
SOCCER TOURNAMENT – IDENTIFYING KEYS

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the starting and ending date of the contract. For each match, we store the referee, with first name, surname (different for each referee), and region of birth; each match has exactly one referee. A match takes place in a stadium, where for each stadium we keep track of its identifying name, capacity and coordinates. Finally, we keep information for the seats of each stadium, namely seat number (unique per stadium), type and booking cost.

SOCCER TOURNAMENT – IDENTIFYING KEYS

For each match, we store the date on which it takes place, the start time, the final score, while also a unique ID. We also store the participating teams, with their name (unique for each team), city origin and the trainer's name. We store the name and the surname of each player (the combination of which can be used to distinguish them) in each team with his date of birth and main position. In addition, we keep track of which players belong (or belonged in the past) to which teams and we store the date their starting and ending date of the contract. For each match, we store the referee, with first name, surname (different for each referee), and region of birth; each match has exactly one referee. A match takes place in a stadium, where for each stadium we keep track of its identifying name, capacity and coordinates. Finally, we keep information for the seats of each stadium, namely seat number (unique per stadium), type and booking cost.

SOCCER TOURNAMENT – STATE AFTER RELATIONSHIP IDENTIFICATION



MATCH (ID, Date, Start Time, Score, Referee, Stadium**)**

PLAYER (First Name, Surname, Birth date, Position**)**

TEAMS_MATCH (Match ID, Team Name**)**

TEAM (Name, City, Trainer**)**

REFEREE (First Name, Surname, Birth Region**)**

PLAYER_TEAM (Player F. Name, Player Surname, Team Name, Start Date, End Date**)**

STADIUM (Name, Capacity, Coordinates**)**

SEAT (Number, Stadium, Type, Cost**)**