Lecture 5: Relational Model - relationships

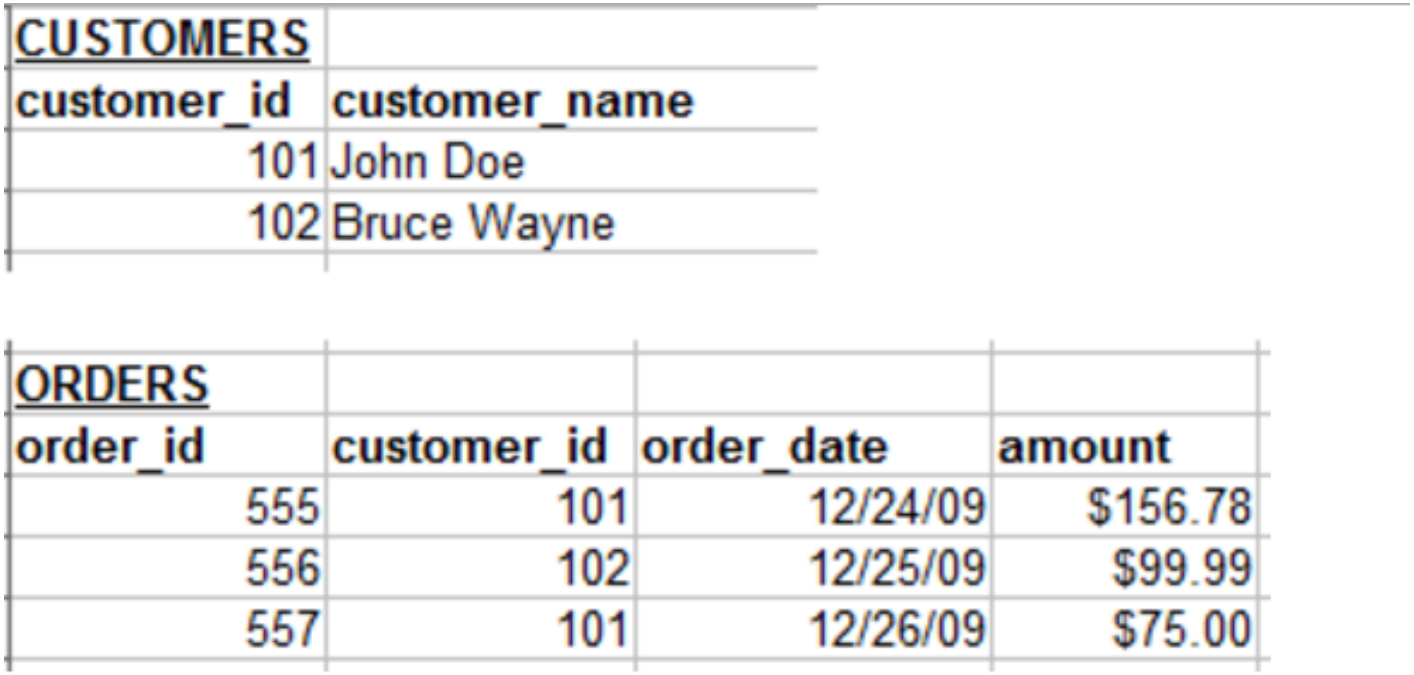
BADM/ACCY 352

Spring 2017

Instructor: Yi Yang, PhD

Review

- Relational Model
  – Primary key, Foreign Key
  – Two integrity rules



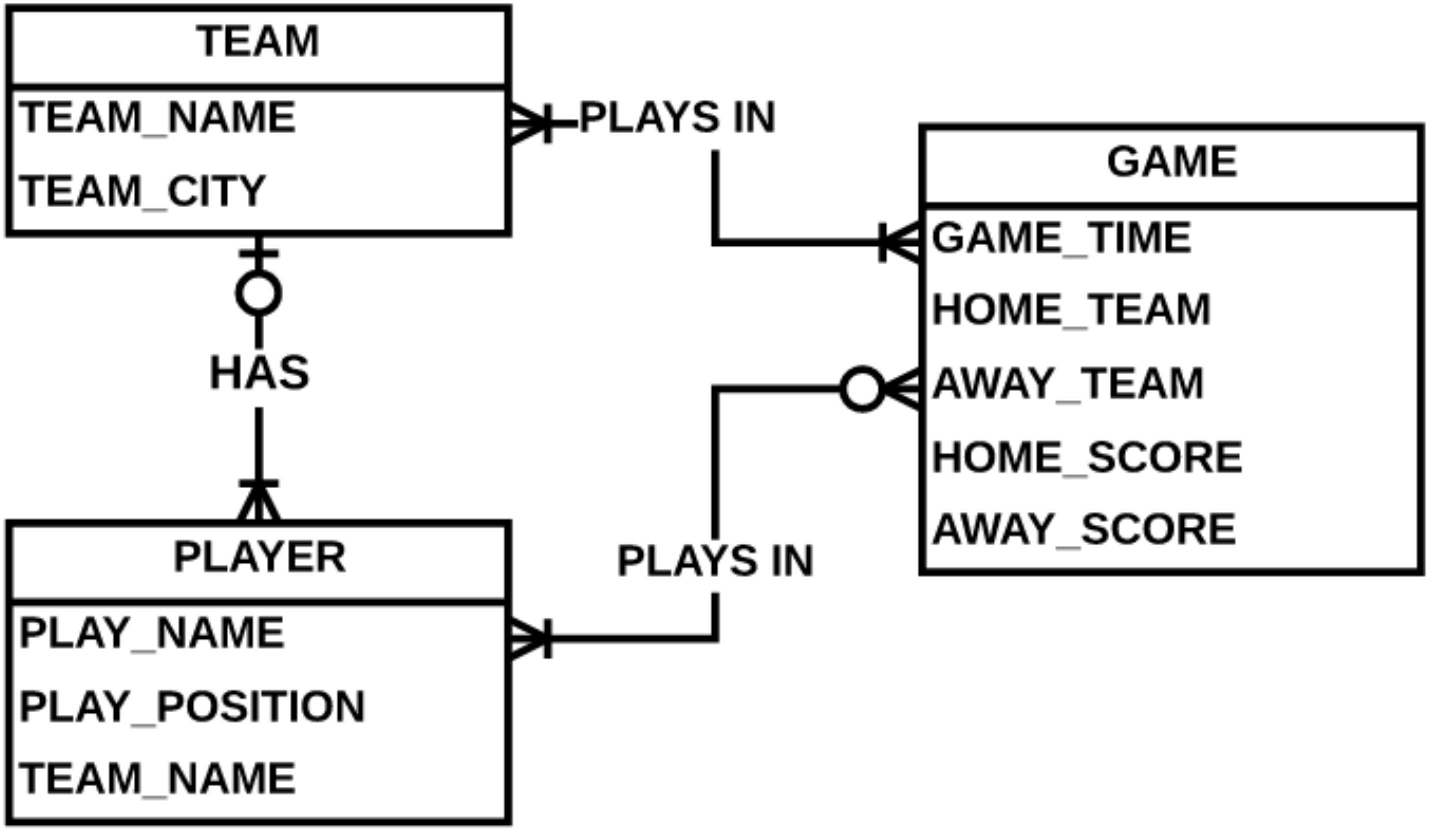
CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne

ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	$156.78
556	102	12/25/09	$99.99
557	101	12/26/09	$75.00

Review

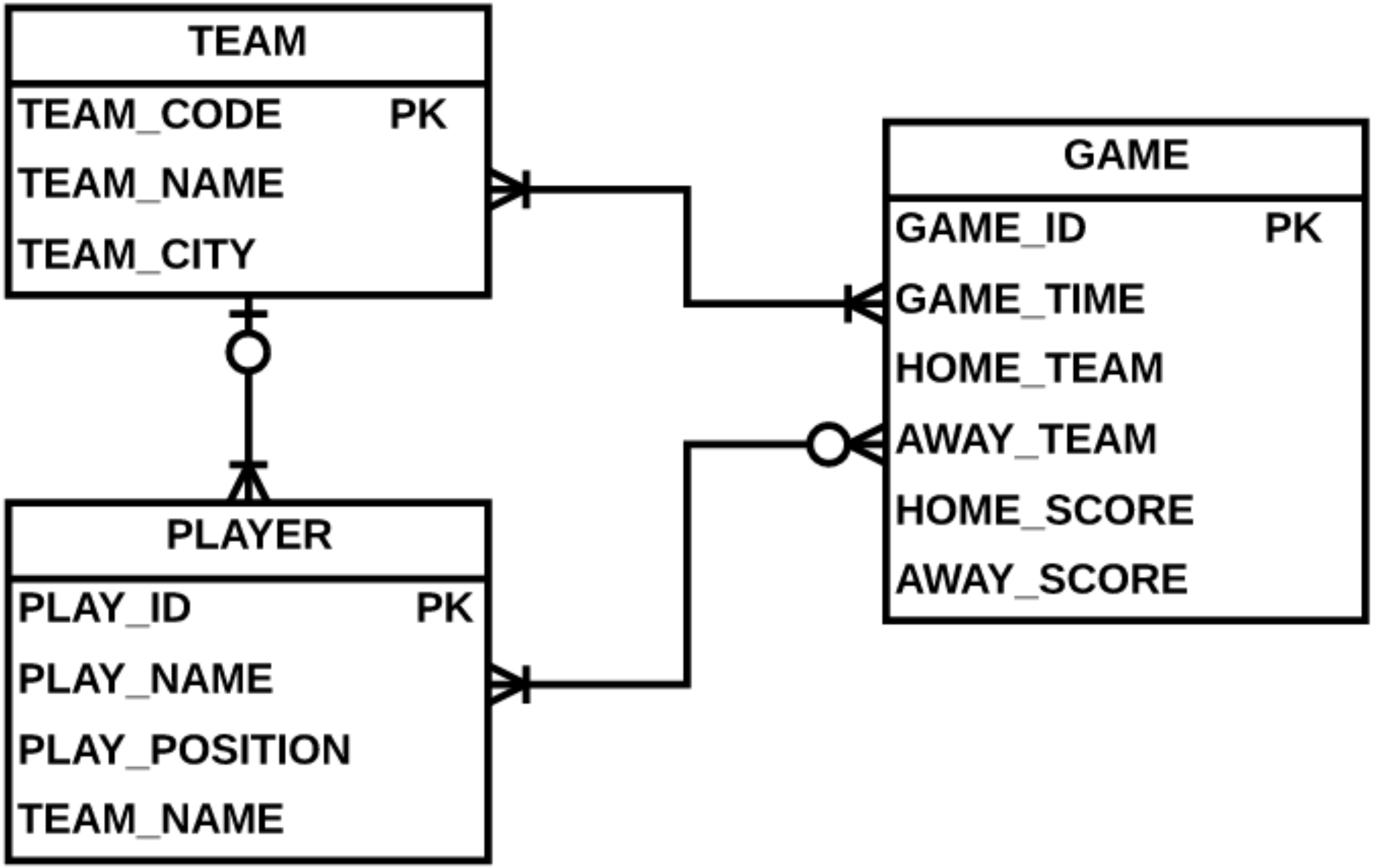
- Relational Model states how the data will be stored in the relational database.
  - Entity -> Table
  - Attribute -> Column
  - Relationship -> ???

- Say you are building a database for 2016 NFL season. The entities are TEAM, PLAYER, GAME.



ER Model

- The first step in relational modeling is to identify PK.
- We still use crow's foot notation.



Relationship

- A relationship in ER Model describes the association between two entities (which are implemented as Tables).
- Relationship in ER Model is implemented when one table has a FK that references the PK of the other table.

One-to-many (1:M) relationship

- In a bank database, a Customer table stores customer data; it also stores customer data in an ACCOUNTS table, which holds information about various bank accounts and associated customers.

- One Customer has Many Accounts.
- One Account is associated with only One Customer.

Implement 1:M relationship

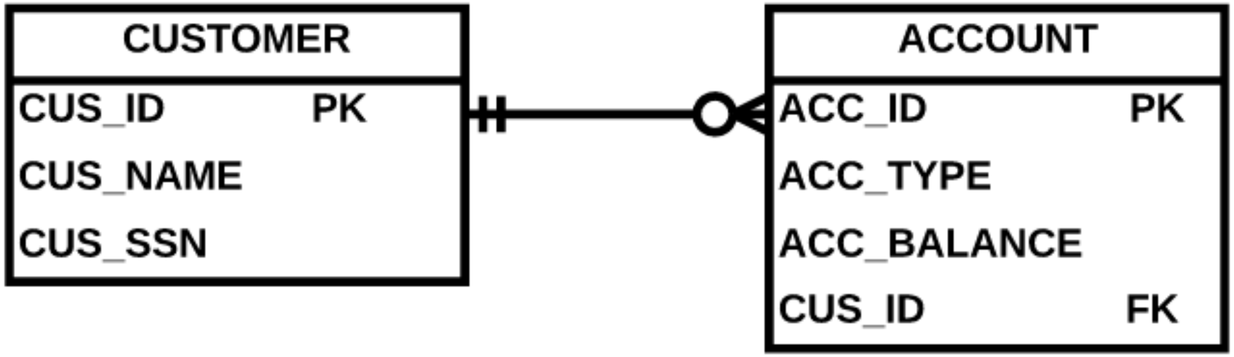
- Put the primary key of the “1” table as the foreign key in the “M” table.

Customer

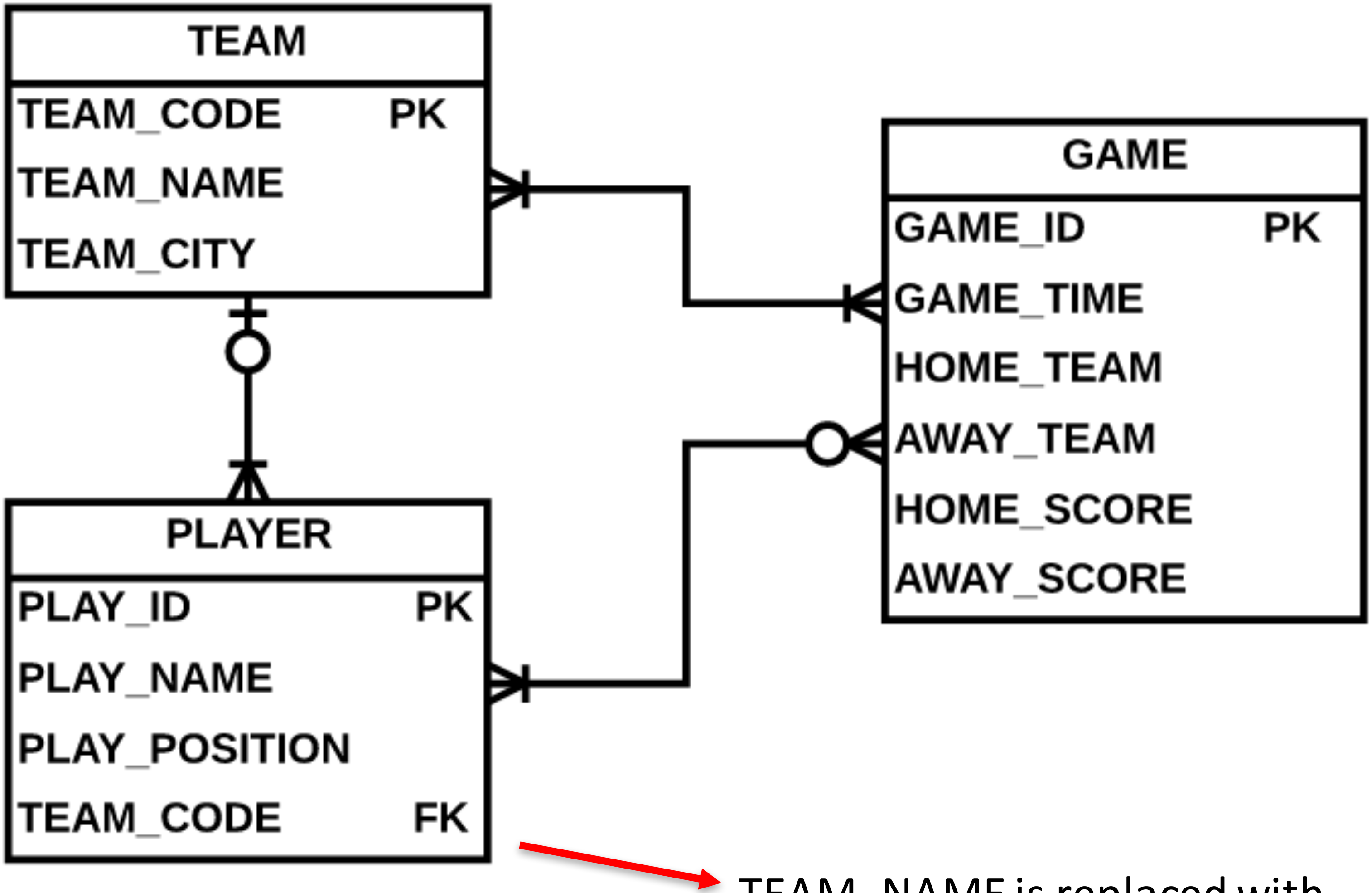
Cus_Id	Cus_Name	Cus_SSN
C001	Alice	123456
C002	Bob	234567
C003	Charlie	345678

Account

Cus_Id	Acc_Num	Acc_Type
C001	012-123	Debit
C001	789-123	Credit
C002	456-123	Credit



Implement 1:M relationship



TEAM
TEAM_CODE PK
TEAM_NAME
TEAM_CITY

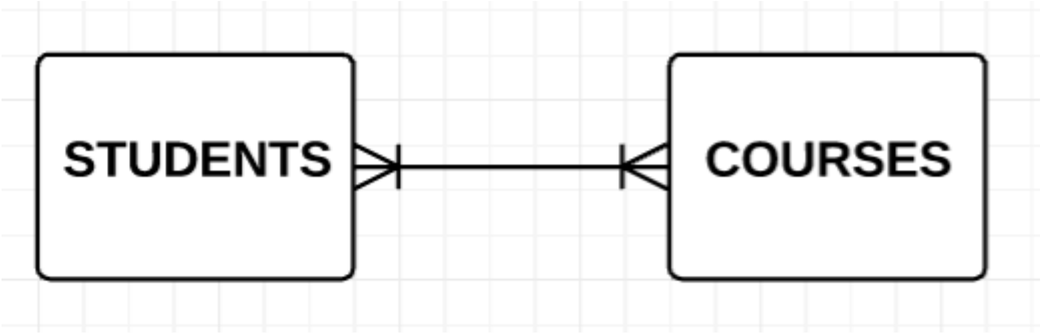
GAME
GAME_ID PK
GAME_TIME
HOME_TEAM
AWAY_TEAM
HOME_SCORE
AWAY_SCORE

PLAYER
PLAY_ID PK
PLAY_NAME
PLAY_POSITION
TEAM_CODE FK

TEAM_NAME is replaced with
TEAM_CODE as FK

Many-to-many (M:N) relationship

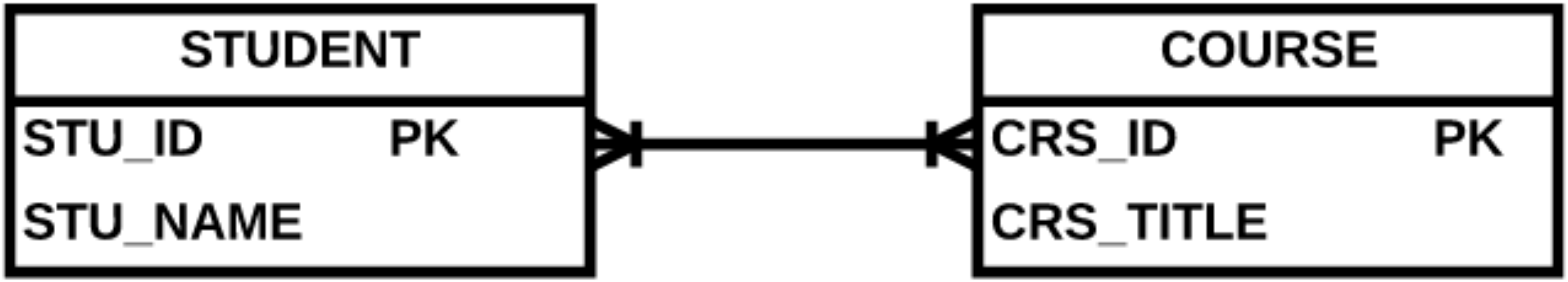
One row in table A is linked with many rows in table B, and vice versa.



A Student can take many courses, and a course can be taken by many students.

Implement M:N relationship

- Where to place the foreign key?

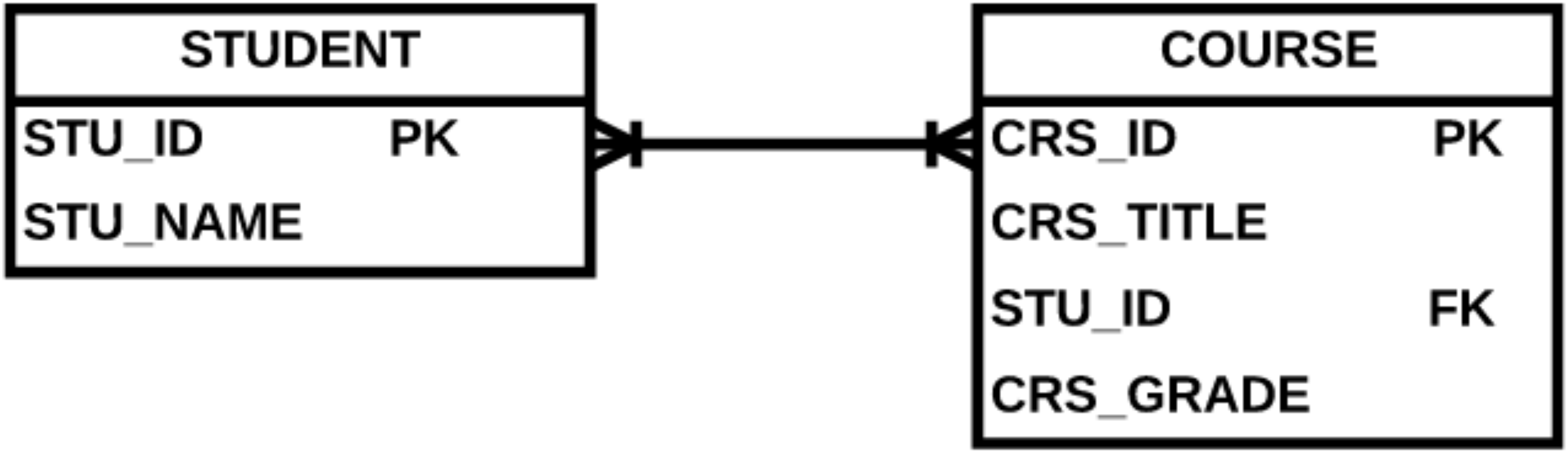


STUDENT	
STU_ID	PK
STU_NAME	

COURSE	
CRS_ID	PK
CRS_TITLE	

- Say Alice takes course Database, Bob takes course Database and Economics.

Case 1: put STUDENT’S PK in COURSE table as FK



STUDENT

STU_ID PK

STU_NAME

COURSE

CRS_ID PK

CRS_TITLE

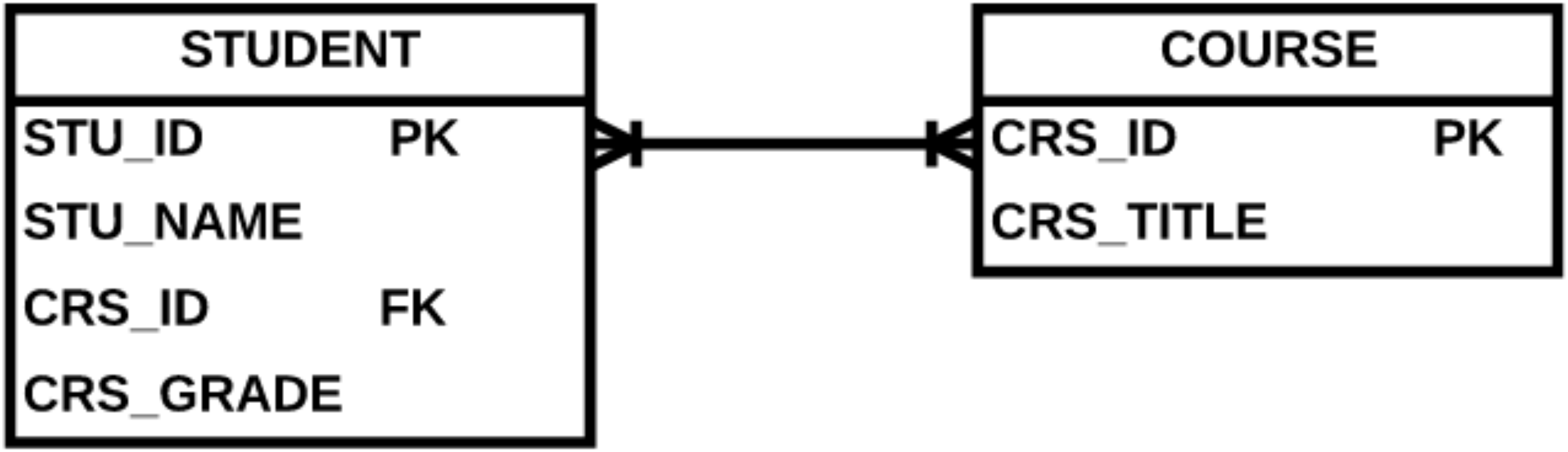
STU_ID FK

CRS_GRADE

STU_ID	STU_NAME
S001	Alice
S002	Bob

STU_ID	COURSE_ID	COURSE_NAME	COURSE_GRADE
S001	C301	Database	87
S002	C301	Database	93
S002	C302	Economics	82

Case 2: put COURSE'S PK in STUDENT table as FK



STUDENT

STU_ID PK

STU_NAME

CRS_ID FK

CRS_GRADE

COURSE

CRS_ID PK

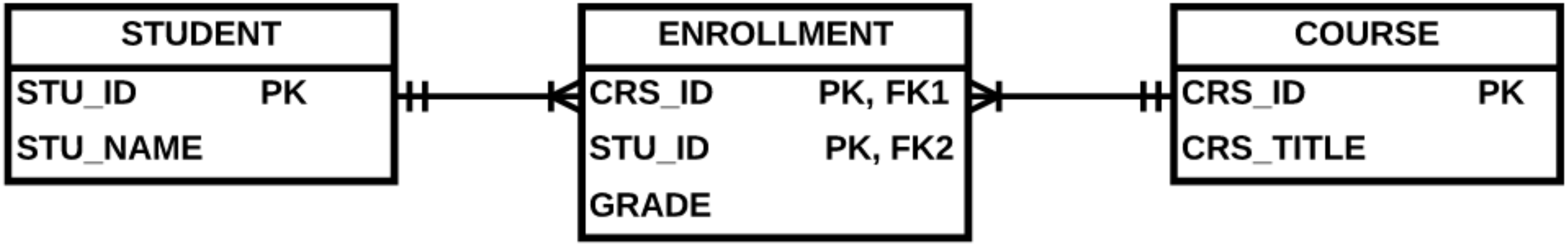
CRS_TITLE

STU_ID	STU_NAME	COURSE_ID	COURSE_GRADE
S001	Alice	C301	87
S002	Bob	C301	93
S002	Bob	C302	82

COURSE_ID	COURSE_NAME
C301	Database
C302	Economics

Implement M:N relationship

- Use a Junction table (composite entity).



STUDENT

STU_ID PK
STU_NAME

ENROLLMENT

CRS_ID PK, FK1
STU_ID PK, FK2
GRADE

COURSE

CRS_ID PK
CRS_TITLE

STU_ID	STU_NAME
S001	Alice
S002	Bob

STU_ID	COURSE_ID	GRADE
S001	C301	87
S002	C301	93
S002	C302	82

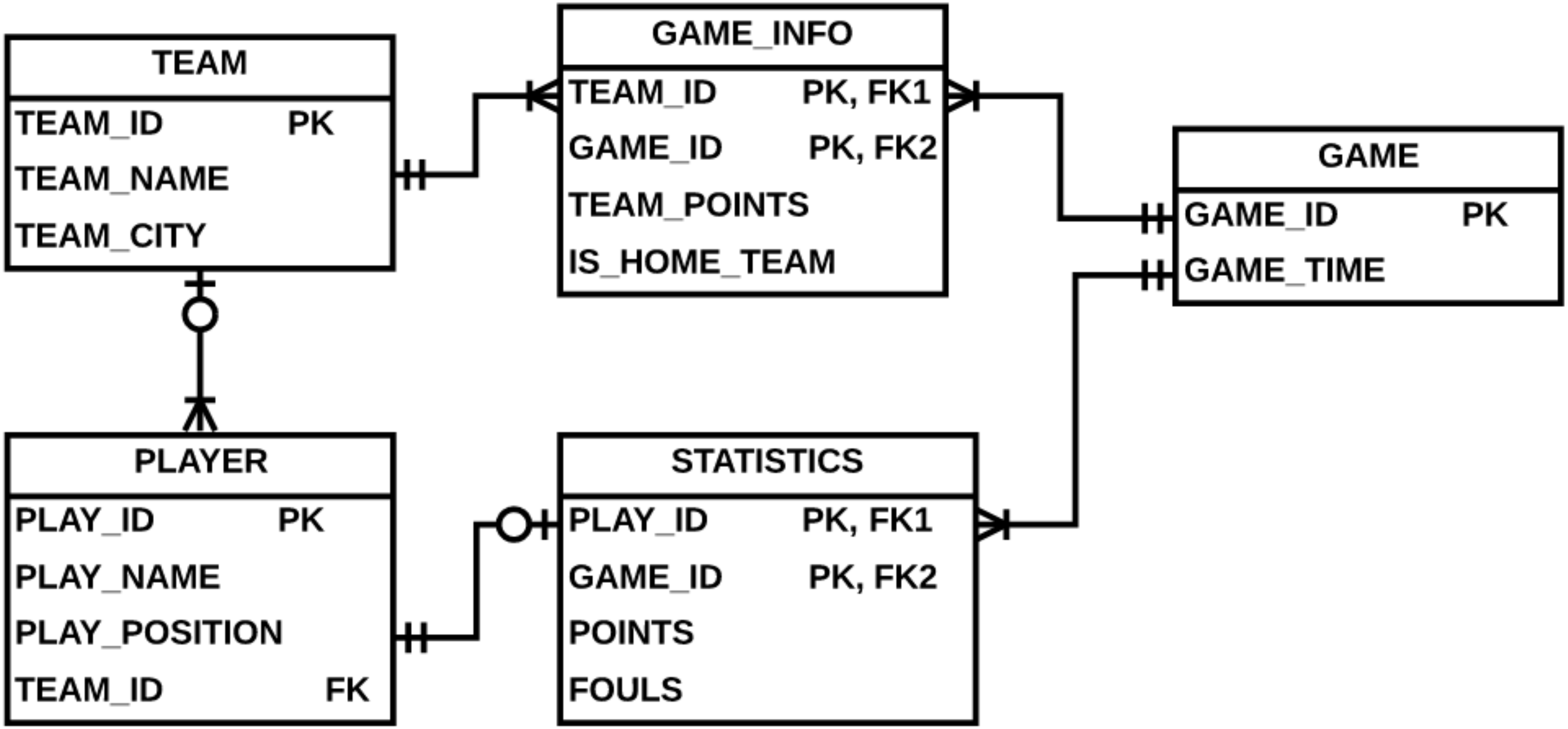
COURSE_ID	COURSE_NAME
C301	Database
C302	Econometrics

Junction table

- What’s the PK of the junction table?
- What’s the FK of the junction table?

- What’s the relationship between the junction table and the original “many” table?
- Break M:N relationship into two 1:M relationships.

Implement M:N relationship



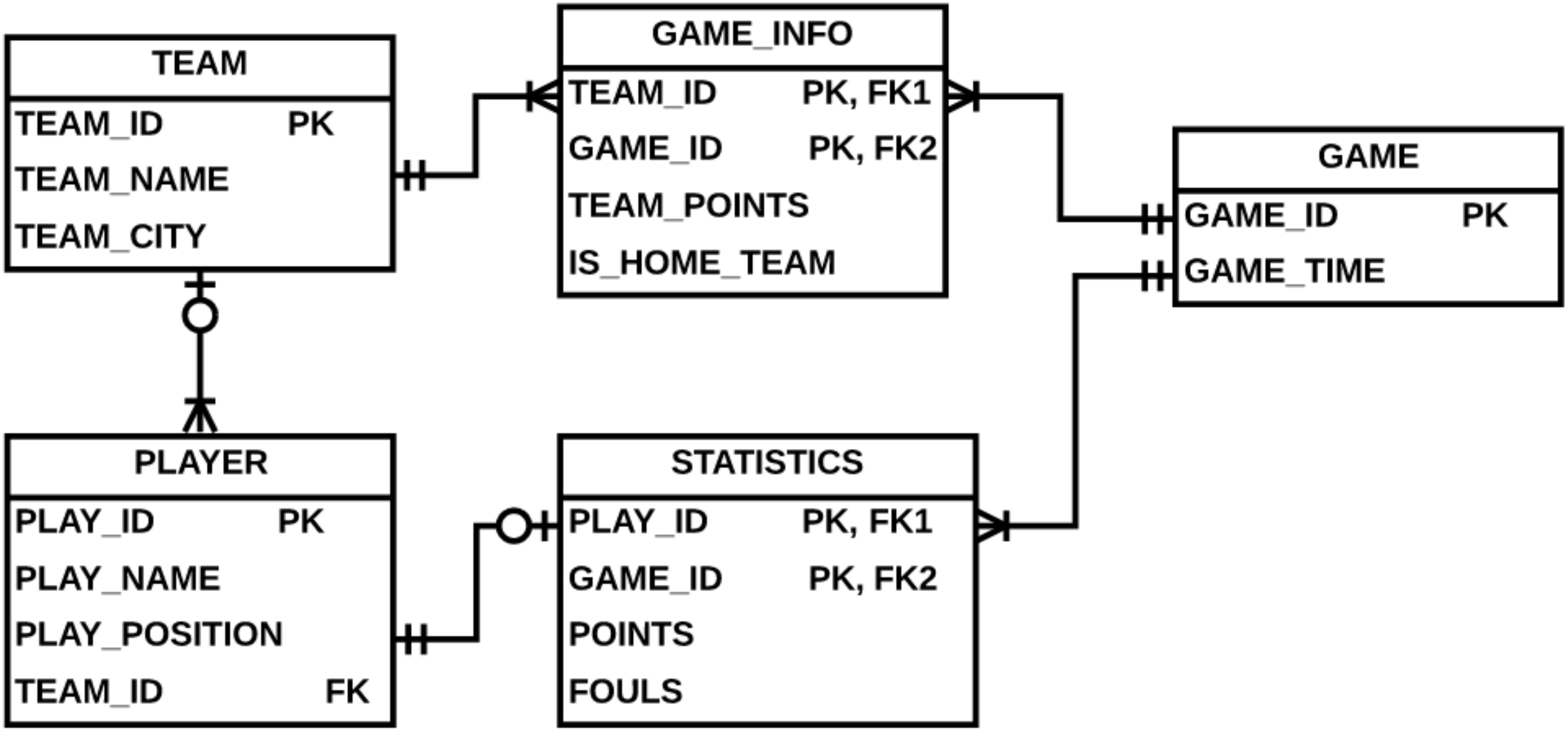
TEAM	
TEAM_ID	PK
TEAM_NAME	
TEAM_CITY	

GAME_INFO	
TEAM_ID	PK, FK1
GAME_ID	PK, FK2
TEAM_POINTS	
IS_HOME_TEAM	

GAME	
GAME_ID	PK
GAME_TIME	

PLAYER	
PLAY_ID	PK
PLAY_NAME	
PLAY_POSITION	
TEAM_ID	FK

STATISTICS	
PLAY_ID	PK, FK1
GAME_ID	PK, FK2
POINTS	
FOULS	



TEAM	
TEAM_ID	PK
TEAM_NAME	
TEAM_CITY	

GAME_INFO	
TEAM_ID	PK, FK1
GAME_ID	PK, FK2
TEAM_POINTS	
IS_HOME_TEAM	

GAME	
GAME_ID	PK
GAME_TIME	

PLAYER	
PLAY_ID	PK
PLAY_NAME	
PLAY_POSITION	
TEAM_ID	FK

STATISTICS	
PLAY_ID	PK, FK1
GAME_ID	PK, FK2
POINTS	
FOULS	

Query this database:

- How many games has Player A played?
- What’s the average points that Player A has scored per game?
- How many home games has Team B played?

1:1 relationship

- Example:
  - A retail company requires that each of its stores be managed by a single employee. Each store manager, who is an employee, managers only a single store.
  - EMPLOYEE manages STORE is 1:1 relationship.

Implement 1:1 relationship

What if we want to implement the 1:1 relationship of EMPLOYEE manages DEPARTMENT? Let's say the manager of HR is Alice, and the manager of IT is Doug.

EMP_ID	EMP_NAME	DEP_CODE
E001	Alice	HR
E002	Bob	HR
E003	Charlie	IT
E004	Doug	IT

DEP_CODE	DEP_NAME
HR	Human Resource
IT	Information Technology

Case 1

EMP_ID	EMP_NAME	DEP_CODE	Manager
E001	Alice	HR	Yes
E002	Bob	HR	
E003	Charlie	IT	
E004	Doug	IT	Yes

DEP_CODE	DEP_NAME
HR	Human Resource
IT	Information Technology

Case 2 ✓

EMP_ID	EMP_NAME	DEP_CODE
E001	Alice	HR
E002	Bob	HR
E003	Charlie	IT
E004	Doug	IT

DEP_CODE	DEP_NAME	EMP_ID
HR	Human Resource	E001
IT	Information Technology	E004

Case 3

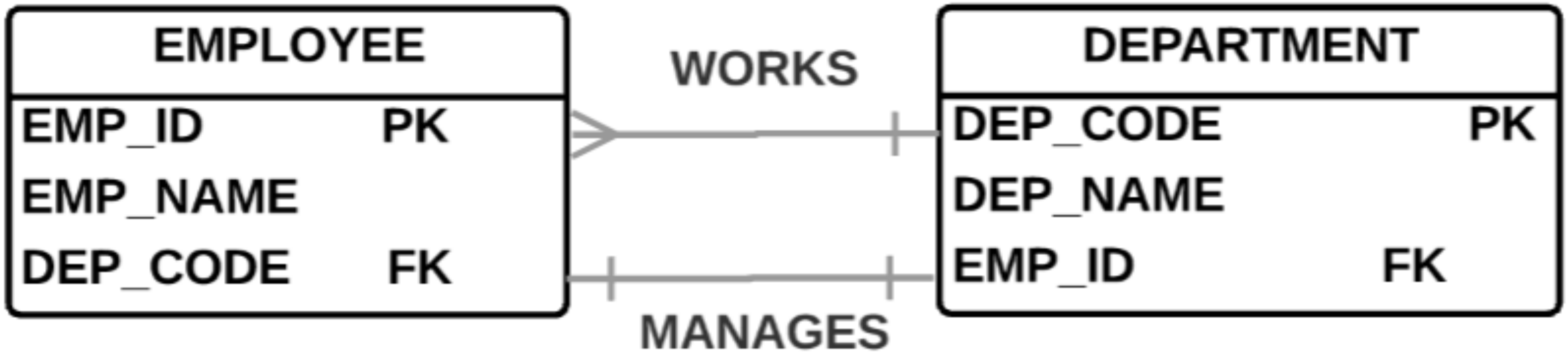
EMP_ID	EMP_NAME	DEP_CODE
E001	Alice	HR
E002	Bob	HR
E003	Charlie	IT
E004	Doug	IT

DEP_CODE	EMP_ID
HR	E001
IT	E004

DEP_CODE	DEP_NAME
HR	Human Resource
IT	Information Technology

1:1 Relationship implementation

- Place the FK that causes the fewest NULLs.



EMPLOYEE

EMP_ID PK
EMP_NAME
DEP_CODE FK

WORKS

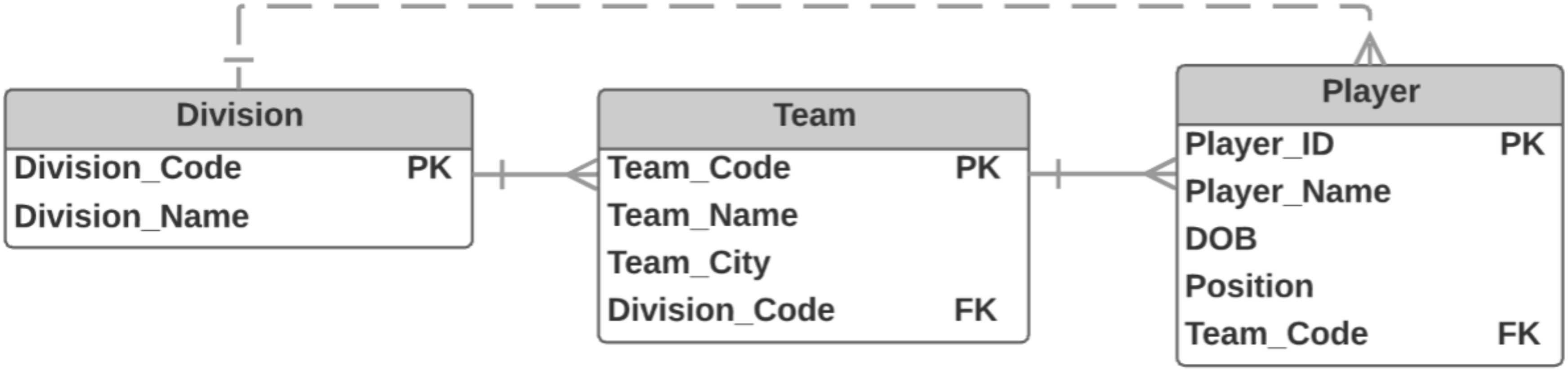
MANAGES

DEPARTMENT

DEP_CODE PK
DEP_NAME
EMP_ID FK

Design case 1: Redundant Relationship

- Redundant relationships occur when there are multiple relationship paths between related entities.
- Design rule: Concise and less redundant



Division	
Division_Code	PK
Division_Name	

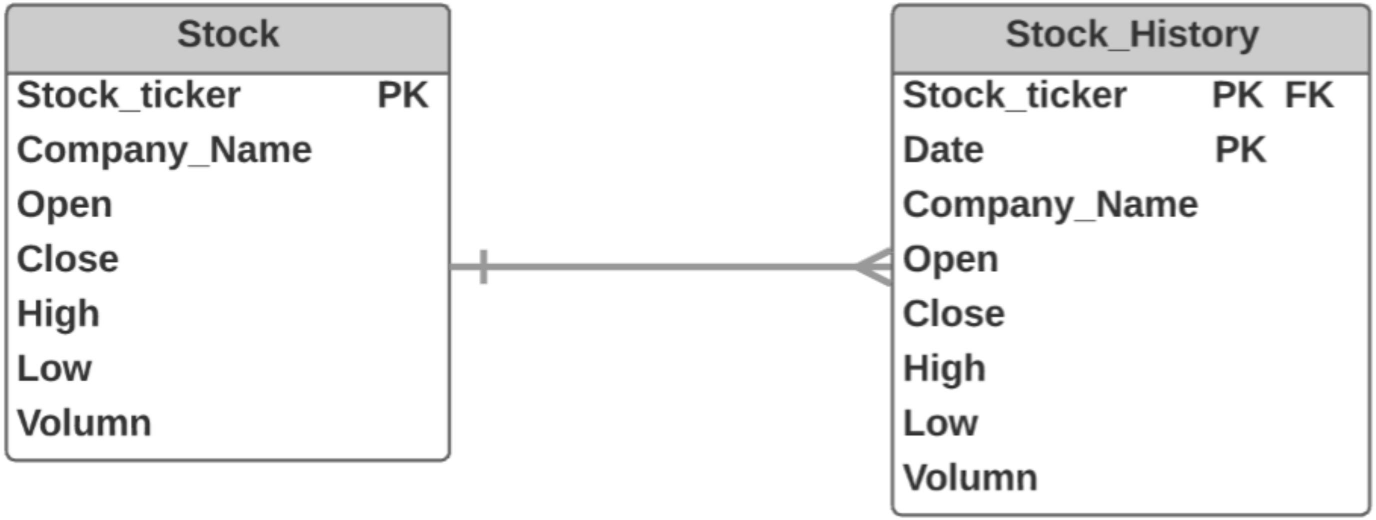
Team	
Team_Code	PK
Team_Name	
Team_City	
Division_Code	FK

Player	
Player_ID	PK
Player_Name	
DOB	
Position	
Team_Code	FK

A redundant relationship

Design case 2: Time-variant Data

- Time-variant data refer to data whose values change over time and for which you ***must*** keep a history of the data changes.
- Attribute value changes: eg. daily stock



Stock

Stock_ticker PK
Company_Name
Open
Close
High
Low
Volumn

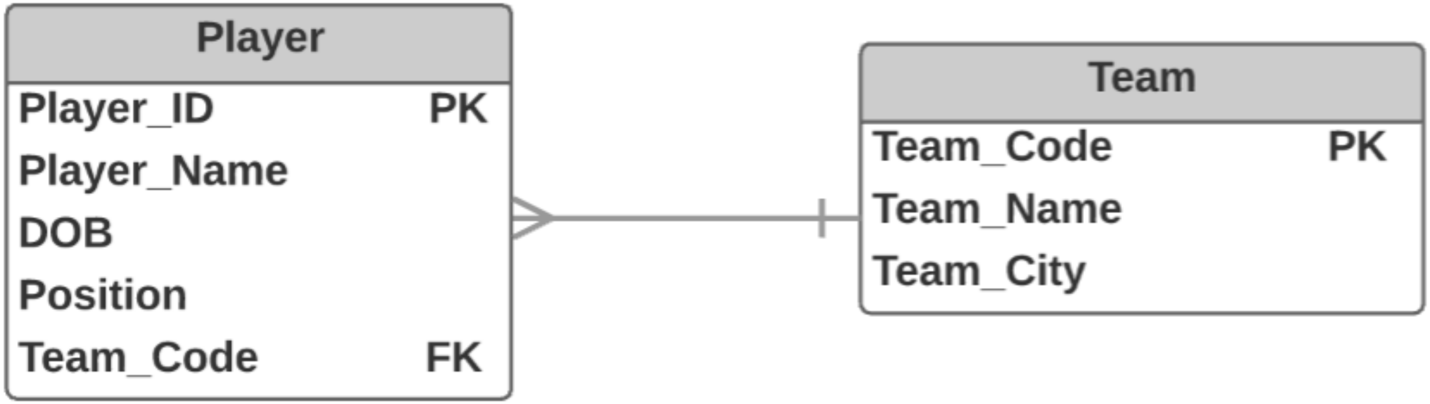
Stock_History

Stock_ticker PK FK
Date PK
Company_Name
Open
Close
High
Low
Volumn

Design case 2: Time-variant Data

- Relationship between entities changes: eg. Player and Team

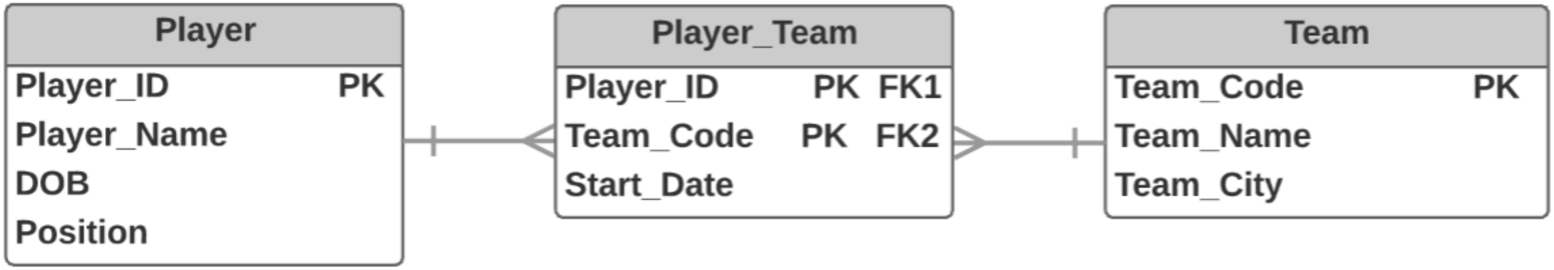
Current



Player	
Player_ID	PK
Player_Name	
DOB	
Position	
Team_Code	FK

Team	
Team_Code	PK
Team_Name	
Team_City	

Historic



Player	
Player_ID	PK
Player_Name	
DOB	
Position	

Player_Team		
Player_ID	PK	FK1
Team_Code	PK	FK2
Start_Date		

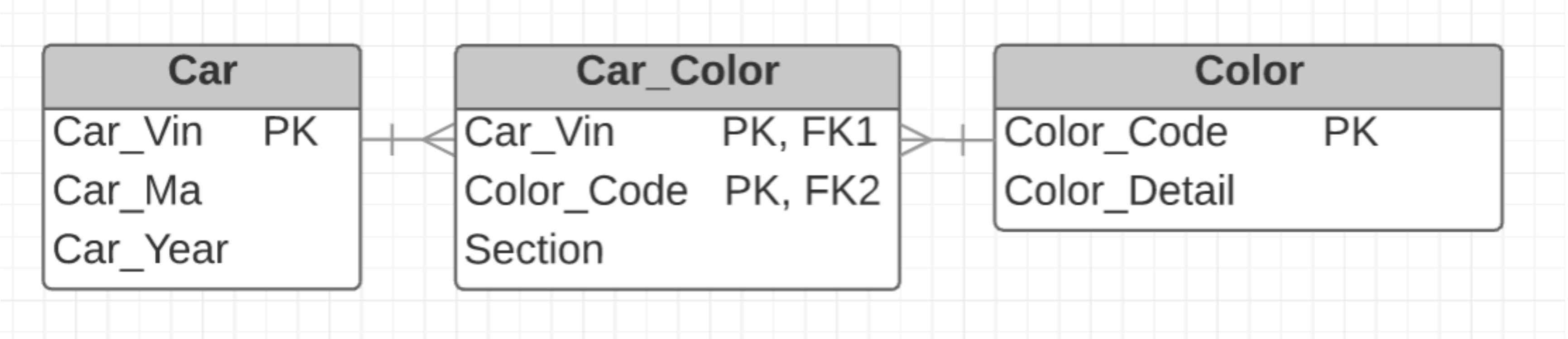
Team	
Team_Code	PK
Team_Name	
Team_City	

Design case 3: multi-value attribute

- An attribute of a record can have multiple values.
- For example, a car can have multiple colors, therefore, Color is a multi-value attribute of Car.

Car_Vin	Car_Manufacture	Car_Year	Car_Color
C0001	Ford	2009	Silver, Black
C0002	Toyota	2010	Red, Black

Design case 3: multi-value attribute



Car	
Car_Vin	PK
Car_Ma	
Car_Year	

Car_Color	
Car_Vin	PK, FK1
Color_Code	PK, FK2
Section	

Color	
Color_Code	PK
Color_Detail	

Car_Vin	Car_Ma	Car_Year
C0001	Ford	2009
C0002	Toyota	2010

Car_Vin	Color_Code
C0001	SL
C0001	BL
C0002	RD
C0002	BL

Color_Code	Color_Detail
SL	Silver
BL	Black
RD	Red

Learning objective

- How to use PK, FK to implement relationships in relational database