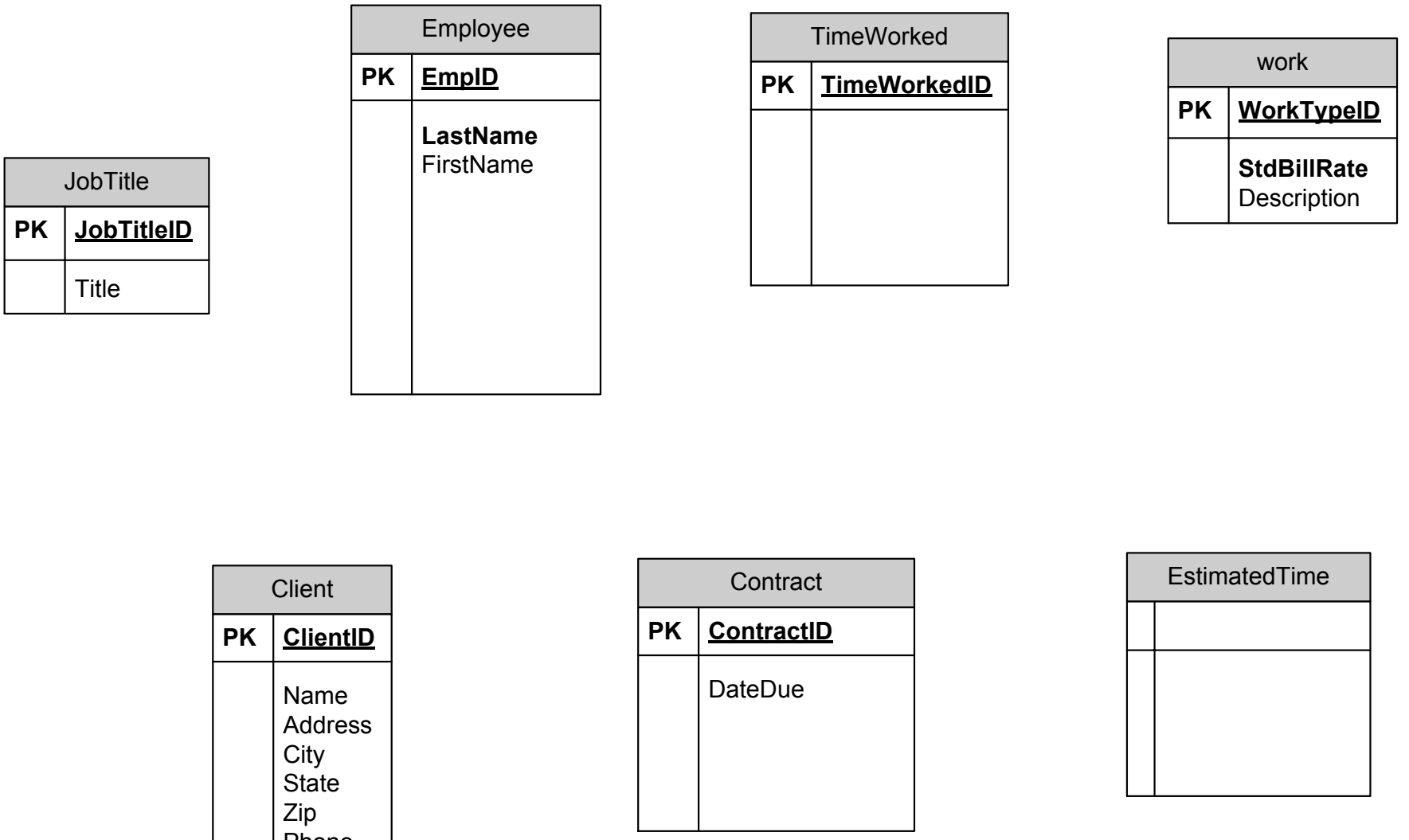
Week 4 Learning Goal and Agenda

Learn how to design different types of relationships

- Binary, ternary, n-ary, unary
- Cardinalities.
- Intersection entities for many-to-many relationships.
- Storing data over time vs. point-in-time.

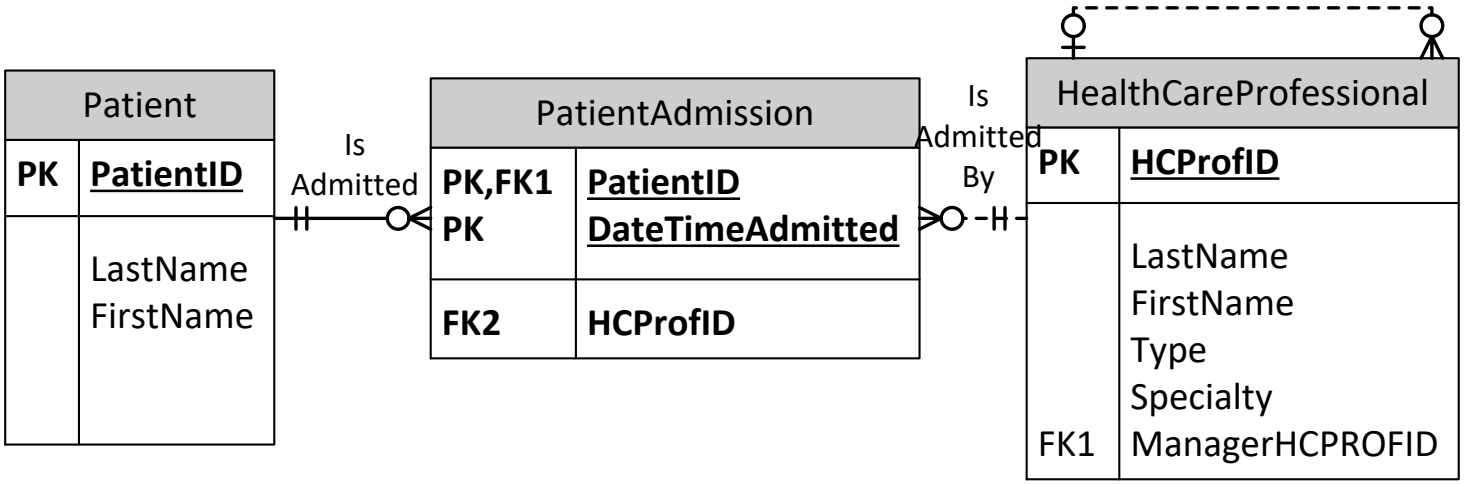
What are we going to do today?

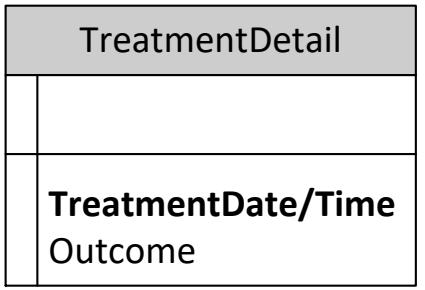
- Review answers to HW#2.
- Finish HCProf Exercise from Week 3.
- Present more complex unary relationships.


JobTitle
PK
JobTitleID
Title
Employee
PK
EmpID
LastName
FirstName
TimeWorked
PK
TimeWorkedID
work
PK
WorkTypeID
StdBillRate
Description
Client
PK
ClientID
Name
Address
City
State
Zip
Phone
Contract
PK
ContractID
DateDue
EstimatedTime

Relationships

- A relationship is an interpretation of a business rule.
- Sometimes the business rule is not always clear, so they can be difficult to interpret and include in a data model.
- Try to provide as much flexibility as possible. Things change in the future and a good data model is flexible and adaptable.


Patient
PK
PatientID
LastName
FirstName
Is Admitted
PatientAdmission
PK,FK1
PatientID
PK
DateTimeAdmitted
FK2
HCProfID
Is Admitted By
HealthCareProfessional
PK
HCProfID
LastName
FirstName
Type
Specialty
FK1
ManagerHCPROFID


TreatmentDetail
TreatmentDate/Time
Outcome

Once admitted, a given patient may be treated by no HCPROF's, but could be treated by multiple HCPROFs. The particular HCPROF who admits a given patient may or may not treat that same patient. A particular HCPROF may treat any number of patients, or may not treat any patients.

Whenever a patient is treated, the hospital records the details of the treatment (TREATMENTDETAIL). Attributes of TREATMENTDETAIL include date/time, and outcome. It is possible that more than one HCPROF participates in the same TREATMENTDETAIL for a patient. For example, it is possible that during a single TREATMENTDETAIL like a consultation about a patient's pain, a Neurologist, a Psychologist and a Nurse Practitioner (all are considered HCPROF's) may participate. The hospital wants to keep track of all HCPROF's who participate in a treatment. For each HCPROF who participates in a TREATMENTDETAIL, the hospital wants to record the notes from the HCPROF and the amount of time that the HCPROF spent on the treatment. For example, if three HCPROF's participate in a single TREATMENTDETAIL for a single patient, the hospital wants to keep track of the notes and time spent individually for each of the three HCPROF's.

Heuristics: Primary Key

- A “natural” primary key is composed of attributes that are already part of the application specification.
- A “surrogate” primary key is created by the database designer for the explicit purpose of being a primary key.
- A surrogate primary key should be numeric and should not contain any of the existing data.
- A surrogate primary key should NEVER be concatenated.
- For logical database design, try and find a natural primary key.

Heuristics: More about the Primary Key

- Never add more attributes than are absolutely necessary to create a primary key value unique.
- Use the primary key of the related strong entity as part of the primary key for the associative entity, if the relationship with the associative entity is mandatory.
- Use the primary keys of the strong entities as part or all of the primary key for the related intersection entity.

Heuristics: Redundant Data

- Avoid redundant data that is composed of long alphanumeric data types.
  - Examples are names, addresses, comments, notes.
  - Standardize any "descriptive" attributes such as categories or types.
- Put sample data in a few rows of each entity so that you can determine whether or not the data will be redundant.
  - If you don't know what data will be stored, ask your client.

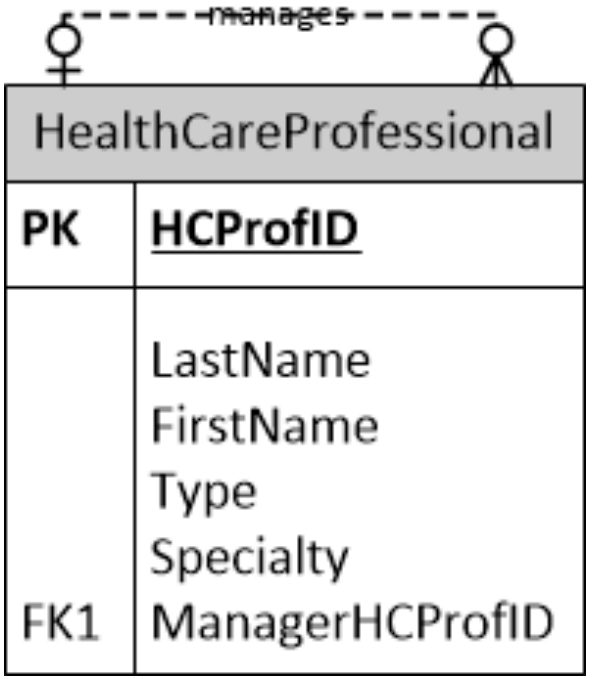
Heuristics: Relationships

- A "relationship" is between entities – cannot have a relationship with a relationship.
- Do not include M:N relationships.
  - M:N relationships always produce data redundancy.
  - Divide M:N relationships with an intersection entity that will create at least two 1:M relationships.
- A single intersection entity can be used to intersect more than two strong entities.
- In a 1:M relationship, the foreign key is placed in the entity on the "many" side of the relationship.
- It is possible to have more than one relationship between two entities.

Heuristics for 1:1 Relationships

- Examine all 1:1 relationships. Determine whether the attributes could all be placed in just one of the two entities eliminating the need for the relationship.
- In a 1:1 relationship, the foreign key can be placed in either entity. Put the foreign key in the entity that seems most “reasonable” and also will result in the fewest number of null values.

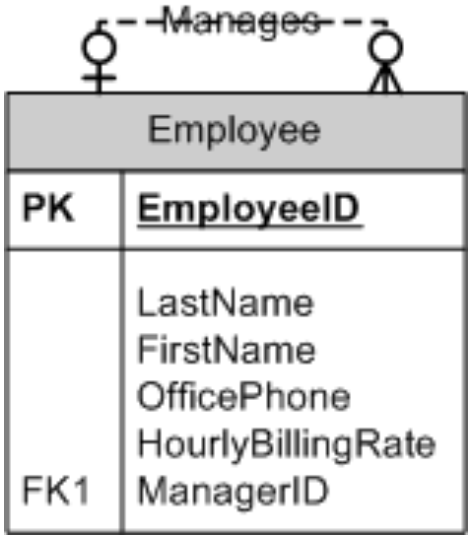
Example of 1:m unary relationship

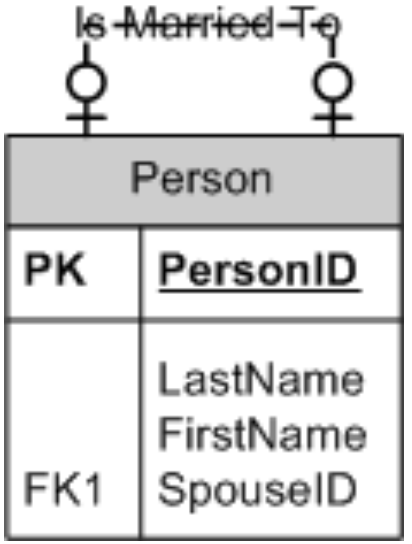


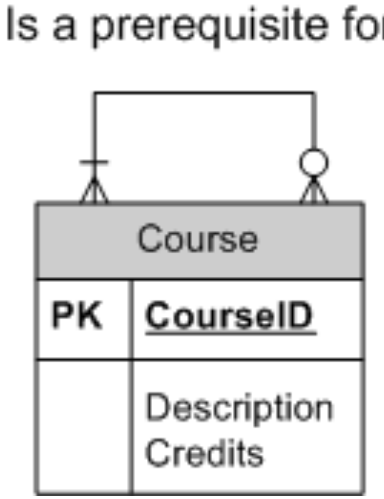
It is possible for row in an entity to have a relationship with another row in the same entity – a "unary" relationship

HCPROFID	LastName	FirstName	Type	Specialty	Manager HCPROFID
10885	Martin	Justine	Physician	Dermatology	45671
92670	Cheng	Randolph	APRN	General	49305
45671	Agarwal	Meera	Administrator	Hospital	Null
49305	Kutty	Mumtaz	Physician	Neurology	45671
40022	Rios	Juan	PA	Geriatrics	45671
57333	Salinas	Marta	APRN	Psychology	49305

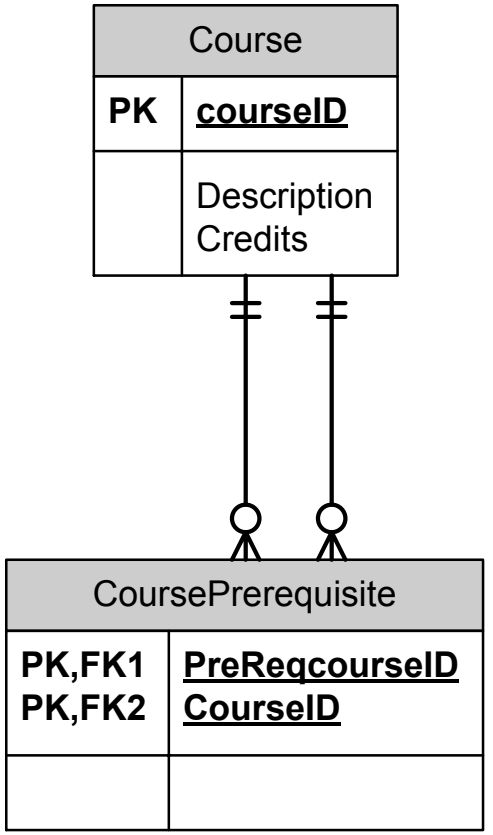
Examples of Unary Relationships







A many-to-many (M:N) unary relationship must be intersected, just like an M:N relationship



CourseID	Description	Credits
IS101	Intro to IS	3
IS201	Computer Applications	3
IS350	Procedural Programming	3
IS475	Database Design	3

CourseID	PrereqCourseID
IS201	IS101
IS475	IS101
IS475	IS201
IS475	IS350
IS350	IS201

IS475/675 Agenda: 02/12/2025

- Review some “rules of thumb” or “heuristics” about database design.
- Present more complex unary relationships.
- Learn how to design a database from data displayed in a form.
- Learn more about modeling data that will be stored over time (longitudinal, time-dependent data).

Heuristics: Primary Key

- A “natural” primary key is composed of attributes that are already part of the application specification.
- A “surrogate” primary key is created by the database designer for the explicit purpose of being a primary key.
- A surrogate primary key should be numeric and should not contain any of the existing data.
- A surrogate primary key should NEVER be concatenated.
- For logical database design, a natural primary key will help a designer understand the database structure.

Heuristics: More about the Primary Key

- Never add more attributes than are absolutely necessary to create a primary key value unique.
- Use the primary key of the related strong entity as part of the primary key for the associative entity, if the relationship with the associative entity is mandatory.
- Use the primary keys of the strong entities as part or all of the primary key for the related intersection entity.

Heuristics: Redundant Data

- Avoid redundant data that is composed of long alphanumeric data types.
  - Example attributes are names, addresses, comments, notes.
  - Example data type is VARCHAR.
  - Standardize any “descriptive” attributes such as categories or types.
- Put sample data in a few rows of each entity so that you can determine whether or not the data will be redundant.
  - If you don’t know what data will be stored, ask your client.

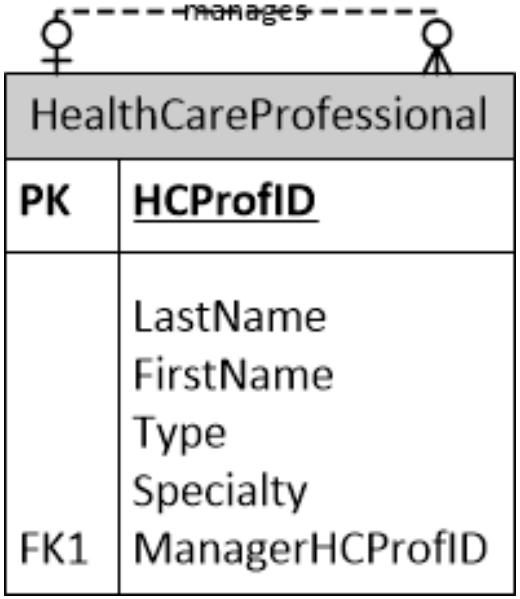
Heuristics: Relationships

- A “relationship” is between entities – cannot have a relationship with a relationship.
- Do not include M:N relationships.
  - M:N relationships always produce data redundancy.
  - Divide M:N relationships with an intersection entity that will create at least two 1:M relationships.
- A single intersection entity can be used to intersect more than two strong entities.
- In a 1:M relationship, the foreign key is placed in the entity on the “many” side of the relationship.
- It is possible to have more than one relationship between two entities.

Heuristics for 1:1 Relationships

- Examine all 1:1 relationships. Determine whether the attributes could all be placed in just one of the two entities eliminating the need for the relationship.
- In a 1:1 relationship, the foreign key can be placed in either entity. Put the foreign key in the entity that seems most “reasonable” and also will result in the fewest number of null values.

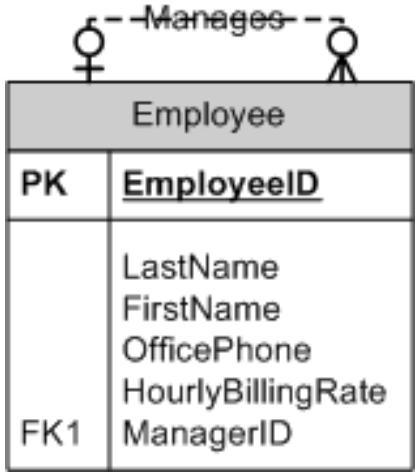
Example of 1:m unary relationship

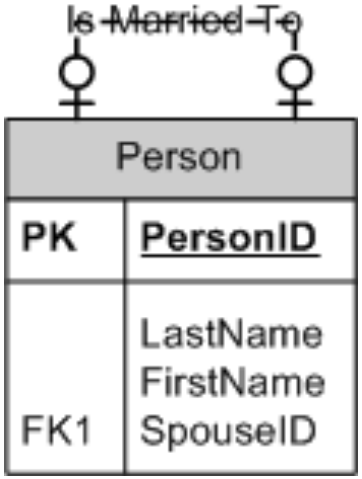


It is possible for row in an entity to have a relationship with another row in the same entity – a "unary" relationship

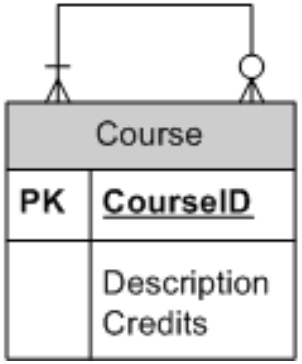
HCPROFID	LastName	FirstName	Type	Specialty	Manager HCPROFID
10885	Martin	Justine	Physician	Dermatology	45671
92670	Cheng	Randolph	APRN	General	49305
45671	Agarwal	Meera	Administrator	Hospital	Null
49305	Kutty	Mumtaz	Physician	Neurology	45671
40022	Rios	Juan	PA	Geriatrics	45671
57333	Salinas	Marta	APRN	Psychology	49305

Examples of Unary Relationships

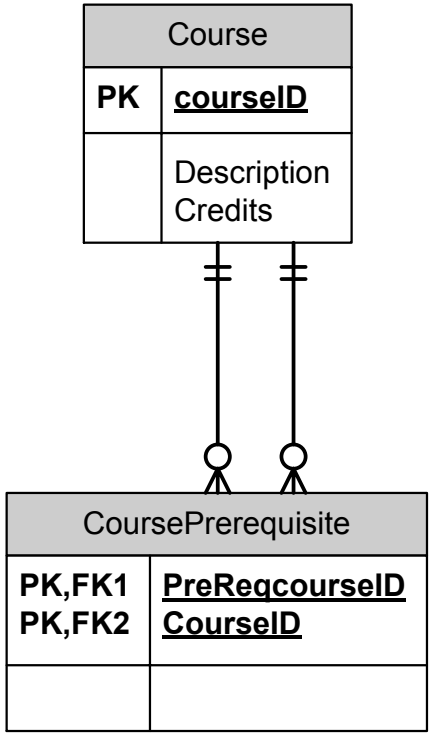




Is a prerequisite for



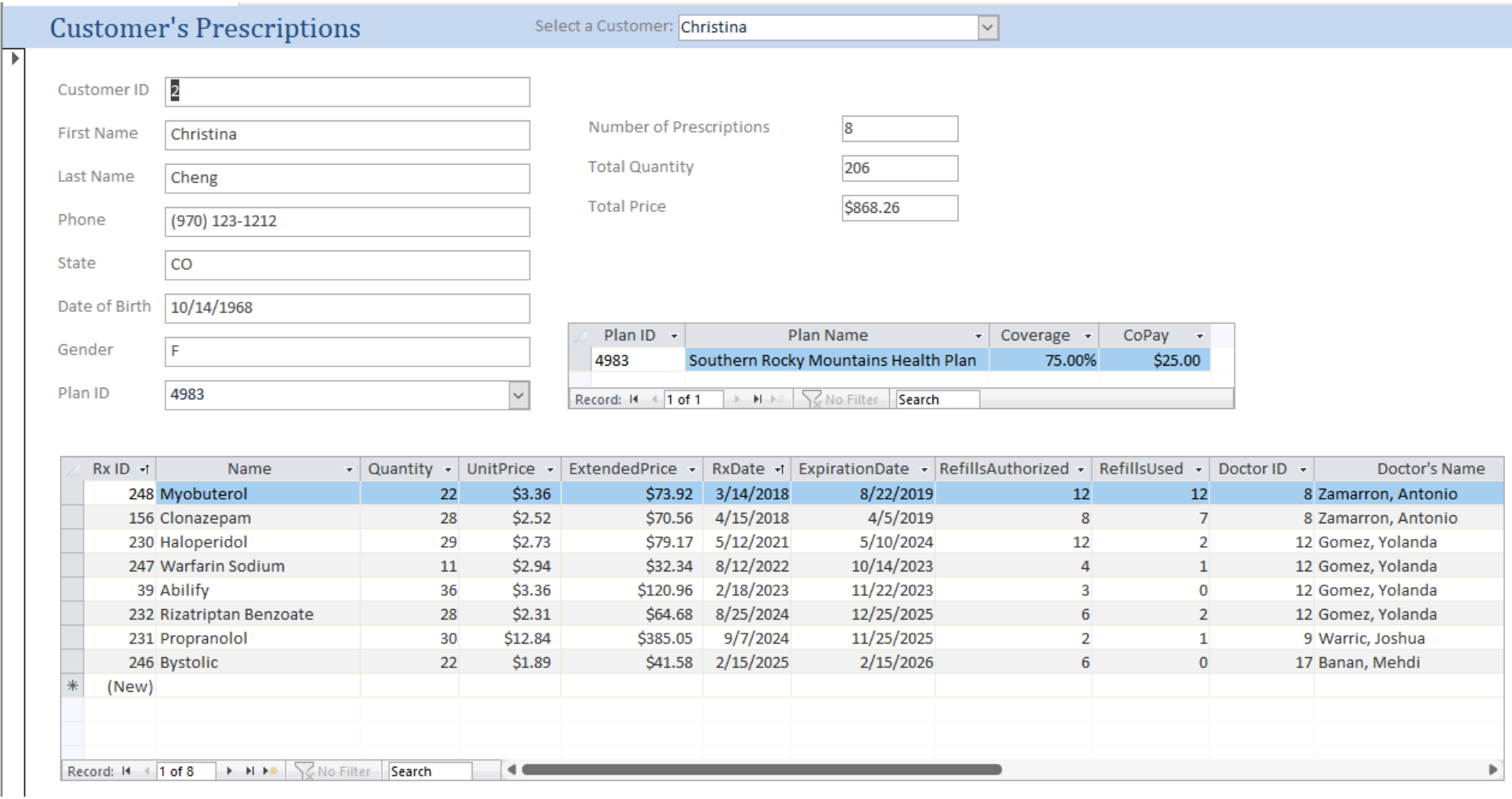
A many-to-many (M:N) unary relationship must be intersected, just like an M:N relationship



CourseID	Description	Credits
IS101	Intro to IS	3
IS201	Computer Applications	3
IS350	Procedural Programming	3
IS475	Database Design	3

CourseID	PrereqCourseID
IS201	IS101
IS475	IS101
IS475	IS201
IS475	IS350
IS350	IS201

Design a database from an input form!



Customer's Prescriptions

Select a Customer: Christina

Customer ID: 2
First Name: Christina
Last Name: Cheng
Phone: (970) 123-1212
State: CO
Date of Birth: 10/14/1968
Gender: F
Plan ID: 4983

Number of Prescriptions: 8
Total Quantity: 206
Total Price: $868.26

Plan ID	Plan Name	Coverage	CoPay
4983	Southern Rocky Mountains Health Plan	75.00%	$25.00

Record: 1 of 1 No Filter Search

Rx ID	Name	Quantity	UnitPrice	ExtendedPrice	RxDate	ExpirationDate	RefillsAuthorized	RefillsUsed	Doctor ID	Doctor's Name
248	Myobuterol	22	$3.36	$73.92	3/14/2018	8/22/2019	12	12	8	Zamarron, Antonio
156	Clonazepam	28	$2.52	$70.56	4/15/2018	4/5/2019	8	7	8	Zamarron, Antonio
230	Haloperidol	29	$2.73	$79.17	5/12/2021	5/10/2024	12	2	12	Gomez, Yolanda
247	Warfarin Sodium	11	$2.94	$32.34	8/12/2022	10/14/2023	4	1	12	Gomez, Yolanda
39	Abilify	36	$3.36	$120.96	2/18/2023	11/22/2023	3	0	12	Gomez, Yolanda
232	Rizatriptan Benzoate	28	$2.31	$64.68	8/25/2024	12/25/2025	6	2	12	Gomez, Yolanda
231	Propranolol	30	$12.84	$385.05	9/7/2024	11/25/2025	2	1	9	Warric, Joshua
246	Bystolic	22	$1.89	$41.58	2/15/2025	2/15/2026	6	0	17	Banan, Mehdi
(New)										

Record: 1 of 8 No Filter Search

Stored data vs. derived data

- Data that can be calculated or determined from other data is not usually stored in a database. This data is called “derived data.”
- Examples of calculations: Counts, totals, average, multiplication.
- Examples of determinations: Class standing (i.e. freshman, sophomore, junior, senior) that is determined based on the number of credits taken by a student.

First Modification

- It is possible that a customer's drug-related health plan will change over time. For example, Christina Cheng is currently using Southern Rocky Mountains Health Plan, but last year she used United Healthcare. Modify your design to keep track of the changes that occur over time in a customer's drug-related health plan.

Cust ID	First Name	Last Name	Address	Date of Birth	Gender	Phone	Health Plan ID	Date Start	Date End
123	Christina	Cheng	CO	10/14/1968	F	910-123-1212	4983	06/01/2024	05/31/2025
125	William	Jonesky	CA	12/17/1988	M	213-233-4712	1228	01/01/2024	12/31/2025
123	Christina	Cheng	CO	10/14/1968	F	910-123-1212	7822	06/01/2023	05/31/2024

Second Modification

- Fred bought more pharmacies. He wants to keep track of the storeID, address and phone number for each store.
- Fred wants to know which of its pharmacy locations originally received a given prescription.

Third Modification

- Each customer can go to any one of the pharmacy locations to fill a prescription. Fred wants to know which of its store locations filled a given prescription.

Last design modification!

The pharmacy wants to keep track of the change in drug unit price over time. The price of a drug can change, and the organization wants to keep track of the start and end dates for the change in unit price for a drug.

Let's "revisit" a problem from HW#2

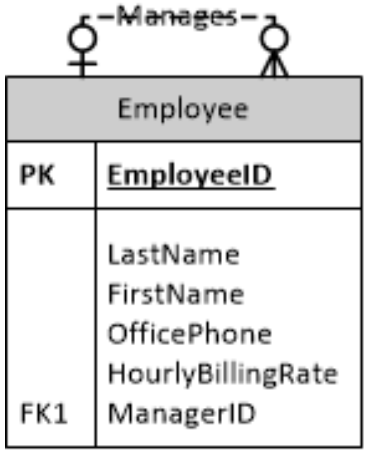
This was an ERD skeleton from HW#2. Design a database to help a consulting organization keep track of the amount of time an employee spends working on a contract. While working each day, each employee completes an online timesheet that looks like this:

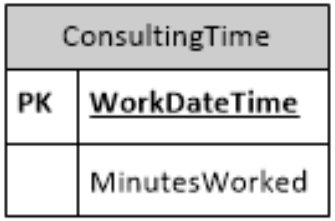
Name	Tristan Elliott	**Date**	2/11/2025
Employee ID	3411		
Contract ID	**Type of Work Description**	**Time Start**	**Minutes Worked**
444	Python Programming	8AM	180
444	Tableau Report Generation	11AM	240
777	Tableau Report Generation	4PM	120

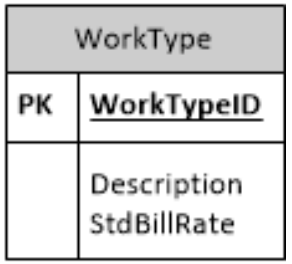
Business rules about the application

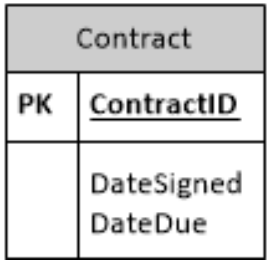
- An employee might work on many contracts, and a contract might have many employees working on it.
- An employee may work on more than one contract during a given day.
- Each time an employee works on a contract, the employee must record a description of the type of work that was done on that contract. The work descriptions are standard across all employees in the organization. Examples of a work description include: "Java Programming," "Tableau Report Generation," "Database Design," and "SQL Programming." An employee may be able to perform many different types of work description, and a type of work description may be able to be performed by many employees.
- An employee has a standard hourly billing rate.
- Each type of work has a standard billing rate.
- The employee's billing rate and the standard billing rate for a type of work do not have to be the same.
- Each employee has only one manager, but a manager may manage multiple employees. An employee does not have to have a manager.
- A contract has only one manager. Each contract must have one manager. A manager manages only one contract at a time.
- Managers are also employees of the consulting organization.
- An employee doesn't have to manage a contract or another employee.
- An employee serves as the salesperson for a contract. The employee who is the manager for a contract is probably not the employee who is the salesperson for that contract.

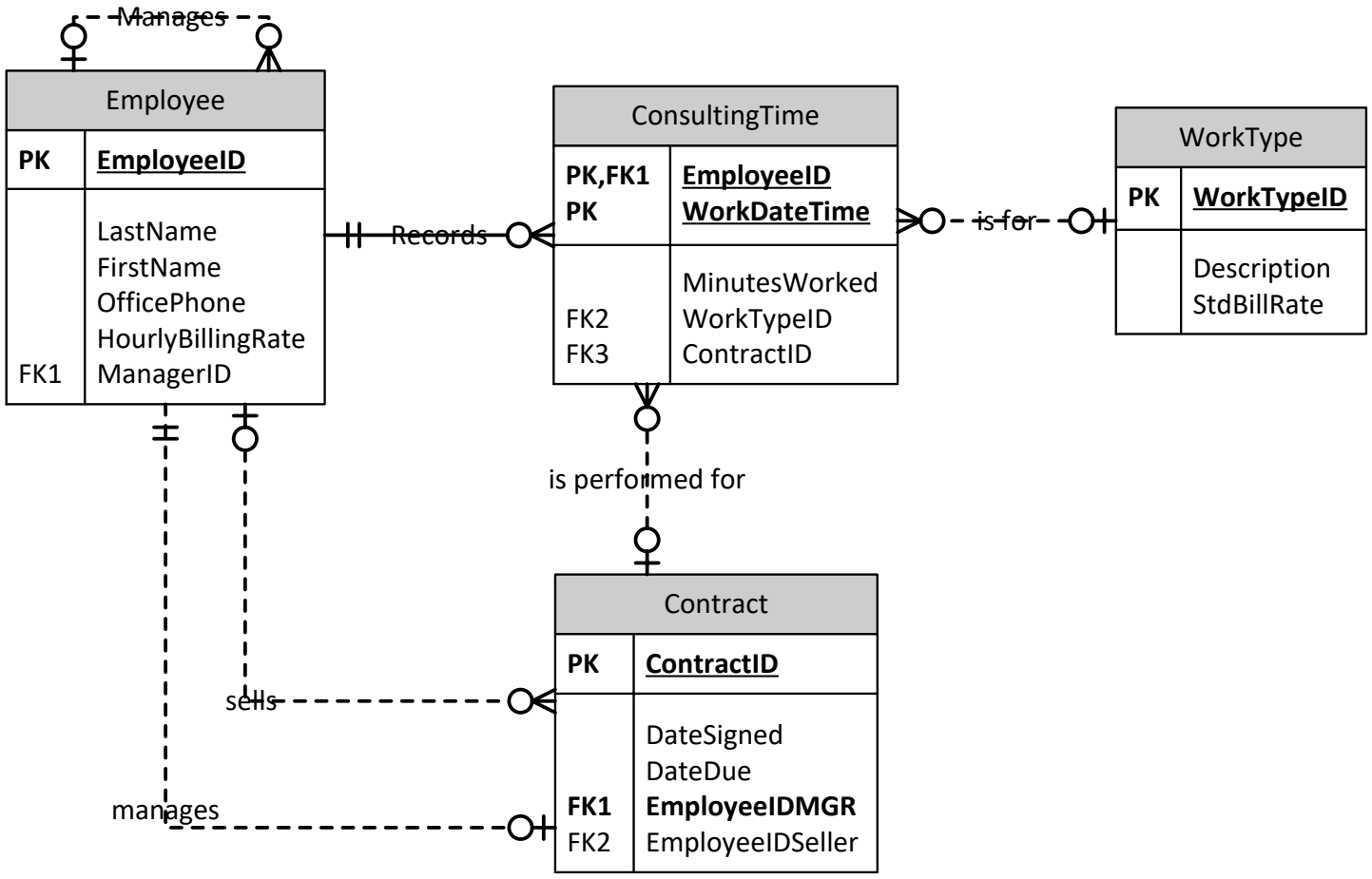
Skeleton ERD










Manages
Employee
PK
EmployeeID
LastName
FirstName
OfficePhone
HourlyBillingRate
FK1
ManagerID
Records
ConsultingTime
PK,FK1
PK
EmployeeID
WorkDateTime
MinutesWorked
FK2
WorkTypeID
FK3
ContractID
is for
WorkType
PK
WorkTypeID
Description
StdBillRate
is performed for
Contract
PK
ContractID
DateSigned
DateDue
FK1
EmployeeIDMGR
FK2
EmployeeIDSeller
sells
manages

What does it mean to store data over time?

- Data that must be maintained beyond a specific point in time.
- Examples:
  - Current data that is updated on an ongoing basis, like a quantity on hand for inventory.
  - Current data that changes, like a person's pay rate.
  - Historical data that must be maintained to create decision-making information.

How long do we store data?

Name	Tristan Elliott	**Date**	1/11/2025
Employee ID	3411		

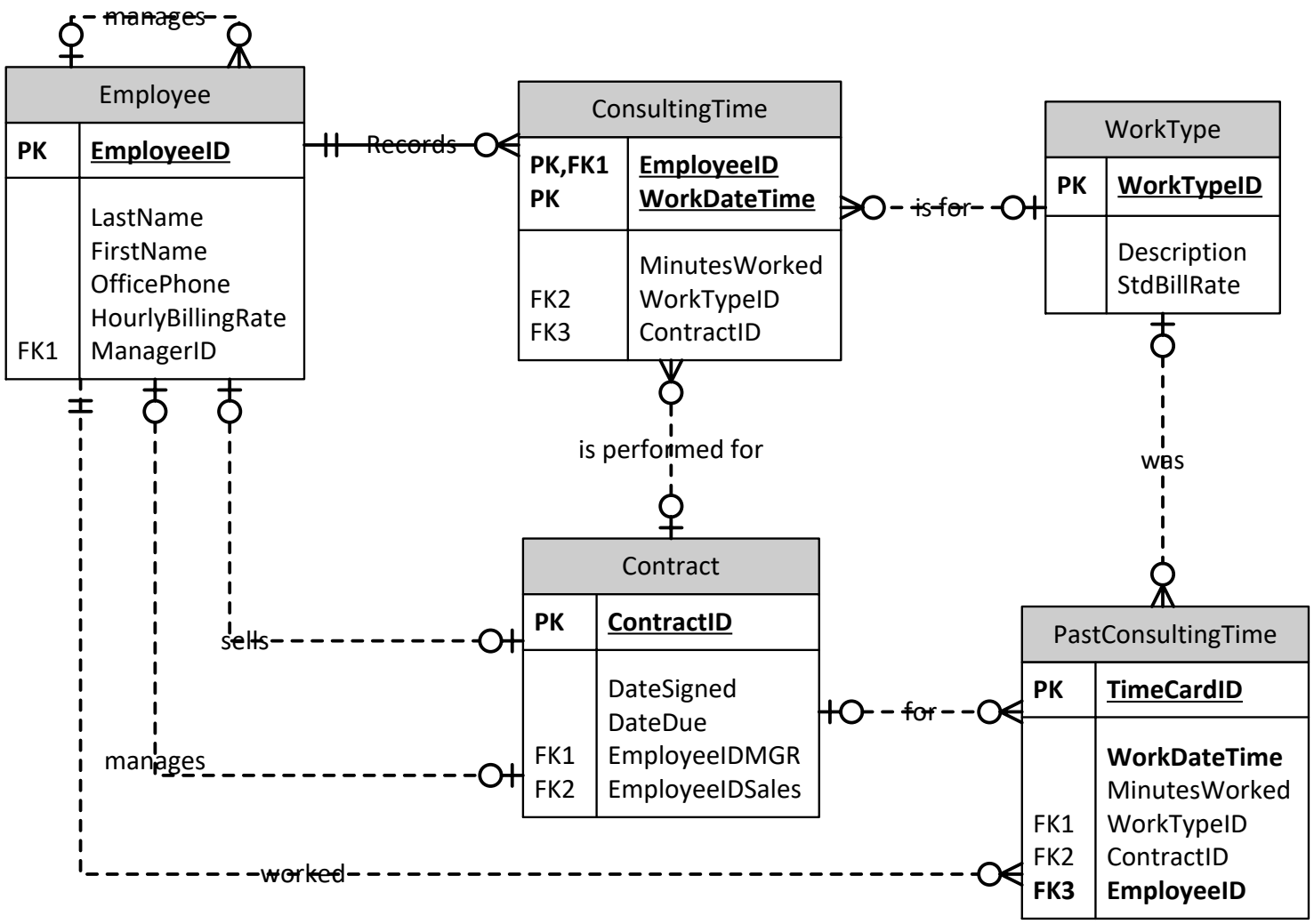
Contract ID	**Type of Work Description**	**Time Start**	**Minutes Worked**
444	Python Programming	8AM	180
444	Tableau Report Generation	11AM	240
777	Tableau Report Generation	4PM	120

What data do we store over time?

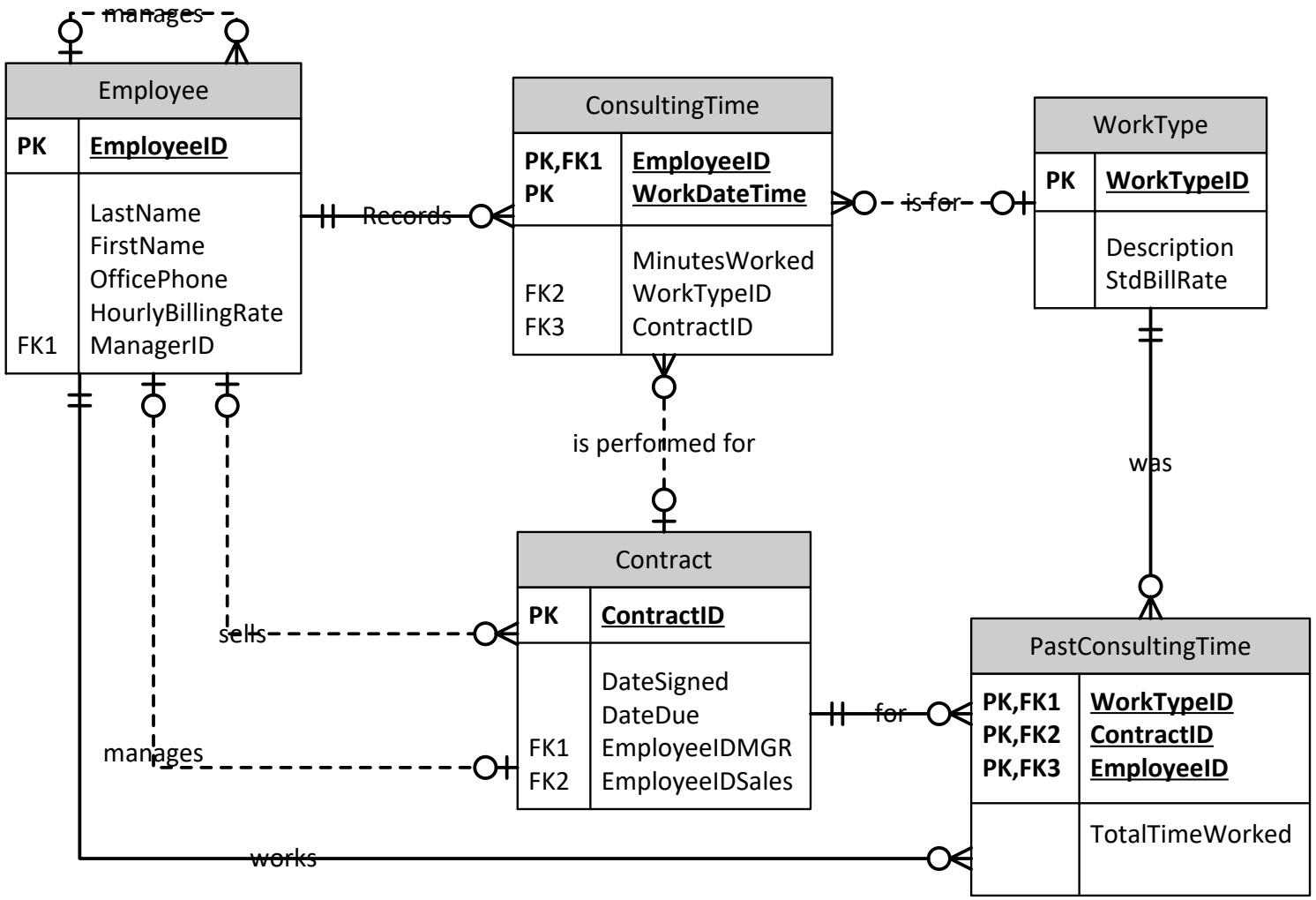
Storing data over time

- Consulting time data is associated with a given date and time (WorkDateTime).
- How long should the data be stored?
- Where should the data be stored long-term, i.e. after the employee is paid or after the contract is completed?
- Should the data be stored at the same level of detail as the original transaction (detailed level of “granularity”) or should it be summarized?

Detailed level of granularity



Summarized level of granularity by Employee, WorkType and Contract



What are the decisions for a database designer?

- Should “old” and “current” data be stored together in a single table?
- Should “old” and “current” data be stored together in a single database?
- What level of granularity is required for the “old” data?
- When should the data be moved from the “current” tables?

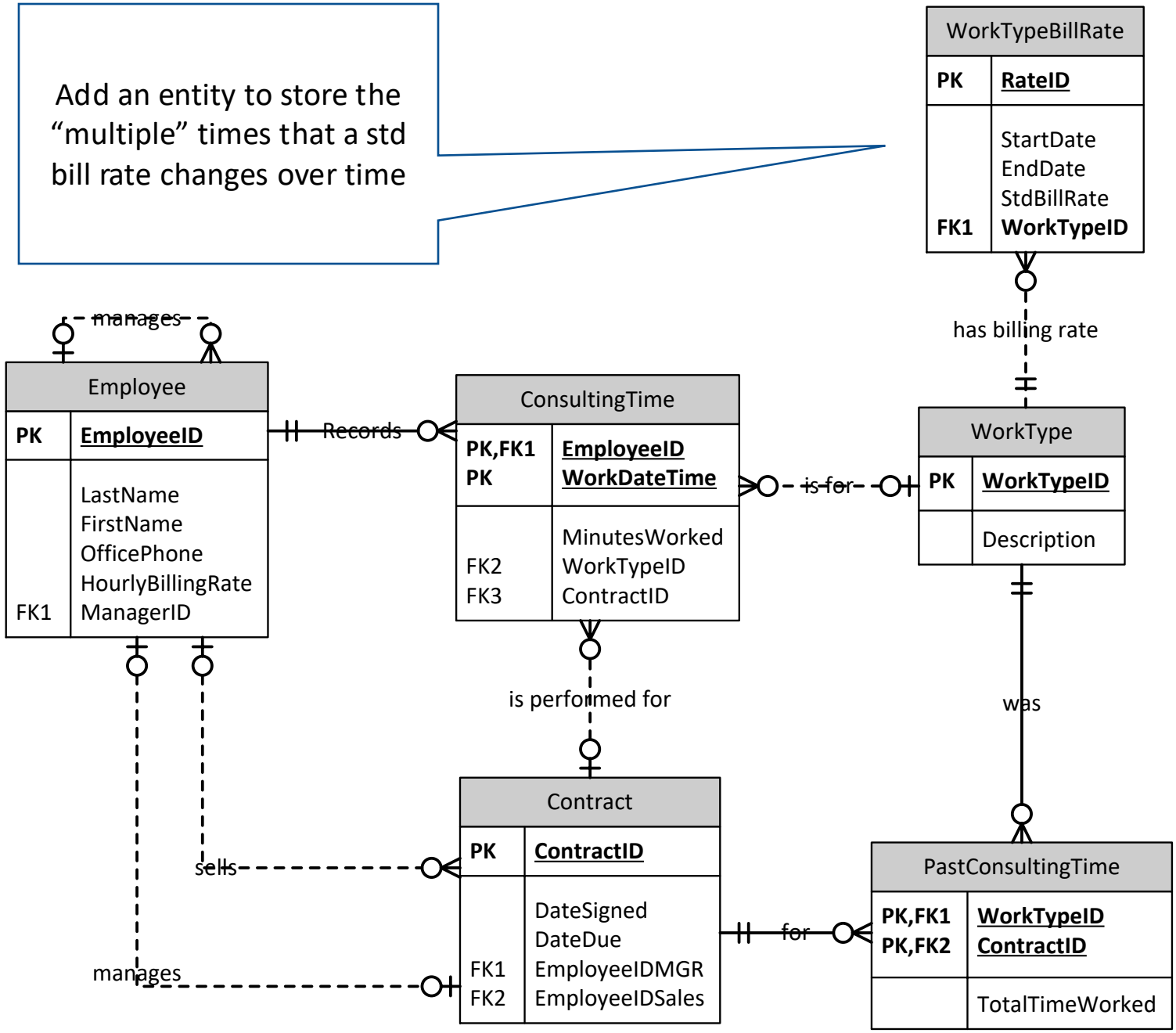
What happens when data changes over time and you want to store the changing data?

- What if you want to keep track of the stdbillrate for a worktype over a period of time?

Time Period	Work Type	Std Bill Rate
1/1/2020 – 8/10/2021	255	$125
8/11/2021 – 12/31/2021	255	$135
1/1/2022 – 12/31/2022	255	$125
1/1/2023 – 01/15/2024	255	$130
1/16/2024– now	255	$175

Slowly changing dimension

- Master data, such as customer or employee data is relatively static.
- Sometimes the data values will change over time. Examples are StdBillRate in the Work Type entity, or HourlyBillingRate in the Employee entity.
- Want to keep a history of the rates rather than overwriting the rates as they change.
- An attribute which changes over time in a master data entity is referred to as a “slowly changing dimension” of that entity.

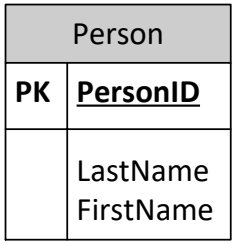

Add an entity to store the "multiple" times that a std bill rate changes over time
WorkTypeBillRate
PK RateID
StartDate
EndDate
StdBillRate
FK1 WorkTypeID
has billing rate
manages
Employee
PK EmployeeID
LastName
FirstName
OfficePhone
HourlyBillingRate
FK1 ManagerID
Records
ConsultingTime
PK,FK1 EmployeeID
PK WorkDateTime
MinutesWorked
FK2 WorkTypeID
FK3 ContractID
is for
WorkType
PK WorkTypeID
Description
is performed for
was
Contract
PK ContractID
DateSigned
DateDue
FK1 EmployeeIDMGR
FK2 EmployeeIDSales
sells
manages
for
PastConsultingTime
PK,FK1 WorkTypeID
PK,FK2 ContractID
TotalTimeWorked

Let's do another design!

Design a database to keep track of the locations where a person has lived. For each person, we want to keep track of his/her personID (unique identifier), last name and first name.

For each person' location, we want to keep track of the start date and end date that the person lived in a place, the address of the location, and the type of location it is (i.e. house, apartment, yurt, condo).

If a person lives in a house, we also want to keep track of the square footage of the house. If a person lives in a multi-family dwelling, we want to keep track of the number of families that could live in that dwelling.


Person
PK
PersonID
LastName
FirstName