IS475/675 Agenda: 02/19/2025

- Review answers to HW#3 – including an ERD evaluation exercise.
- Discuss more about time-dependent data.
- Do time-dependent design exercise.

Let’s do an ERD evaluation exercise!

- Post-sales product registration system.
- Keeps track of registrations.
- Keeps track of reviews.
- Strong entities for customer, toy, product model, distributor.
- Weak entities for registration and review.

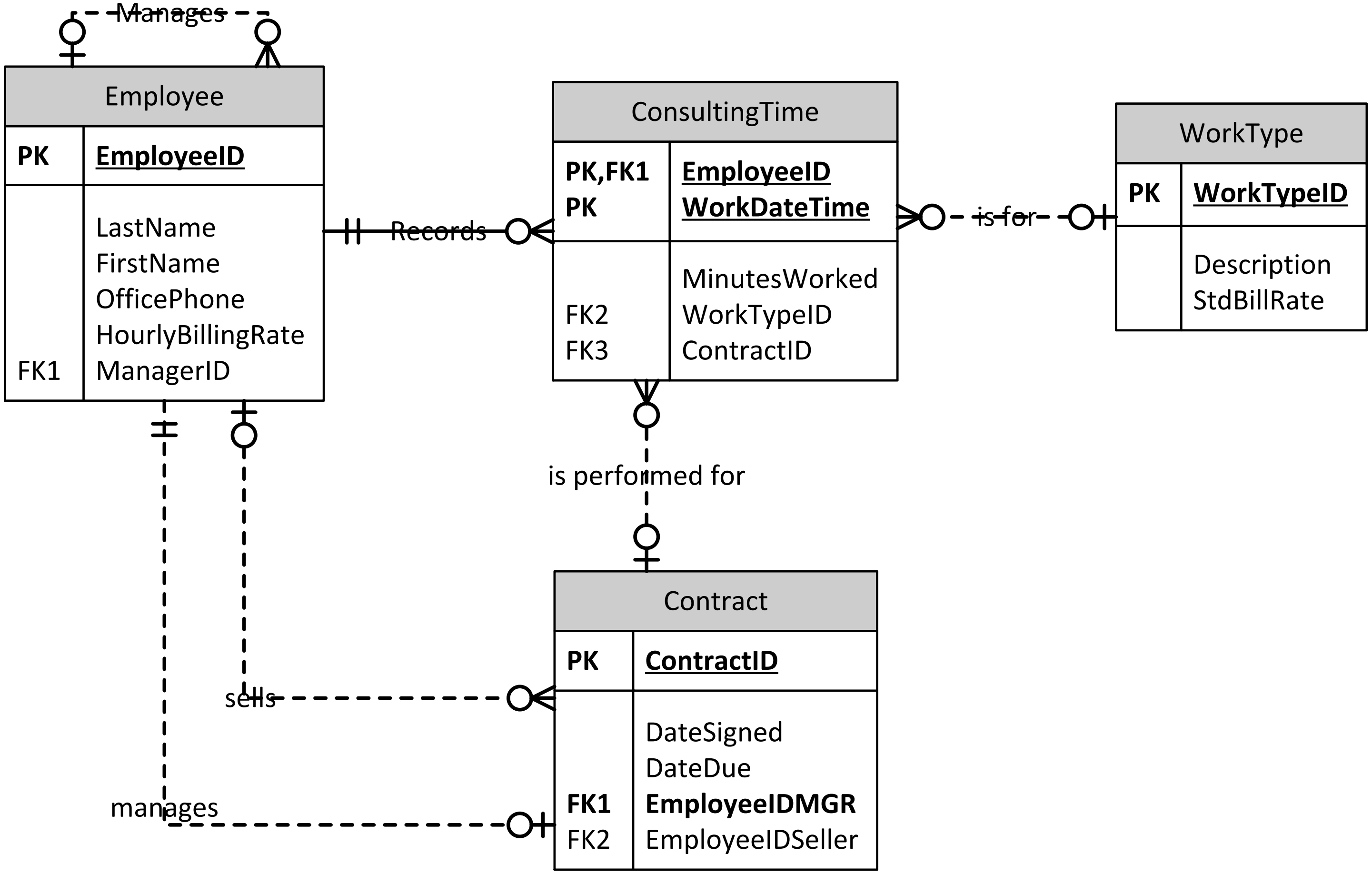
Refresh: what is time-dependent data?

- Data that must be maintained beyond a specific point in time. Stored for compliance or for decision-making.
- Examples:
  - Historical data that must be maintained to produce decision-making information.
  - 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 or a subscription type or an employer.

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**	2/14/2025
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



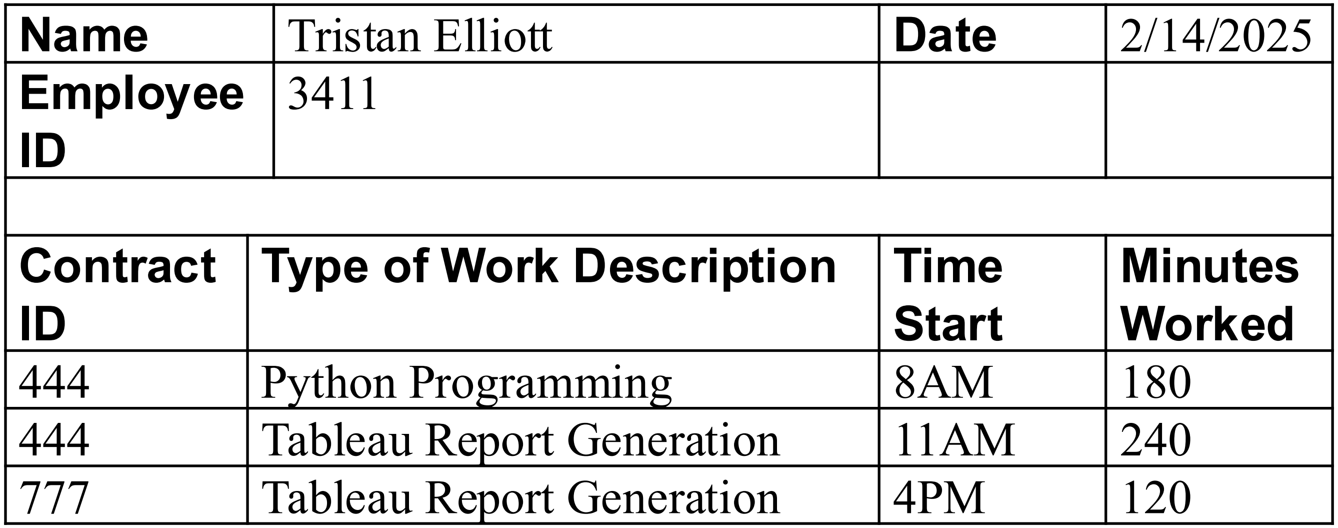
Employee	
PK	**EmployeeID**
    FK1	LastName FirstName OfficePhone HourlyBillingRate ManagerID

ConsultingTime	
PK,FK1 **PK**	**EmployeeID** **WorkDateTime**
 FK2 FK3	MinutesWorked WorkTypeID ContractID

WorkType	
PK	**WorkTypeID**
	Description StdBillRate

Contract	
PK	**ContractID**
  FK1 FK2	DateSigned DateDue **EmployeeIDMGR** EmployeeIDSeller

How long do we store data?



Name	Tristan Elliott	**Date**	2/14/2025
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

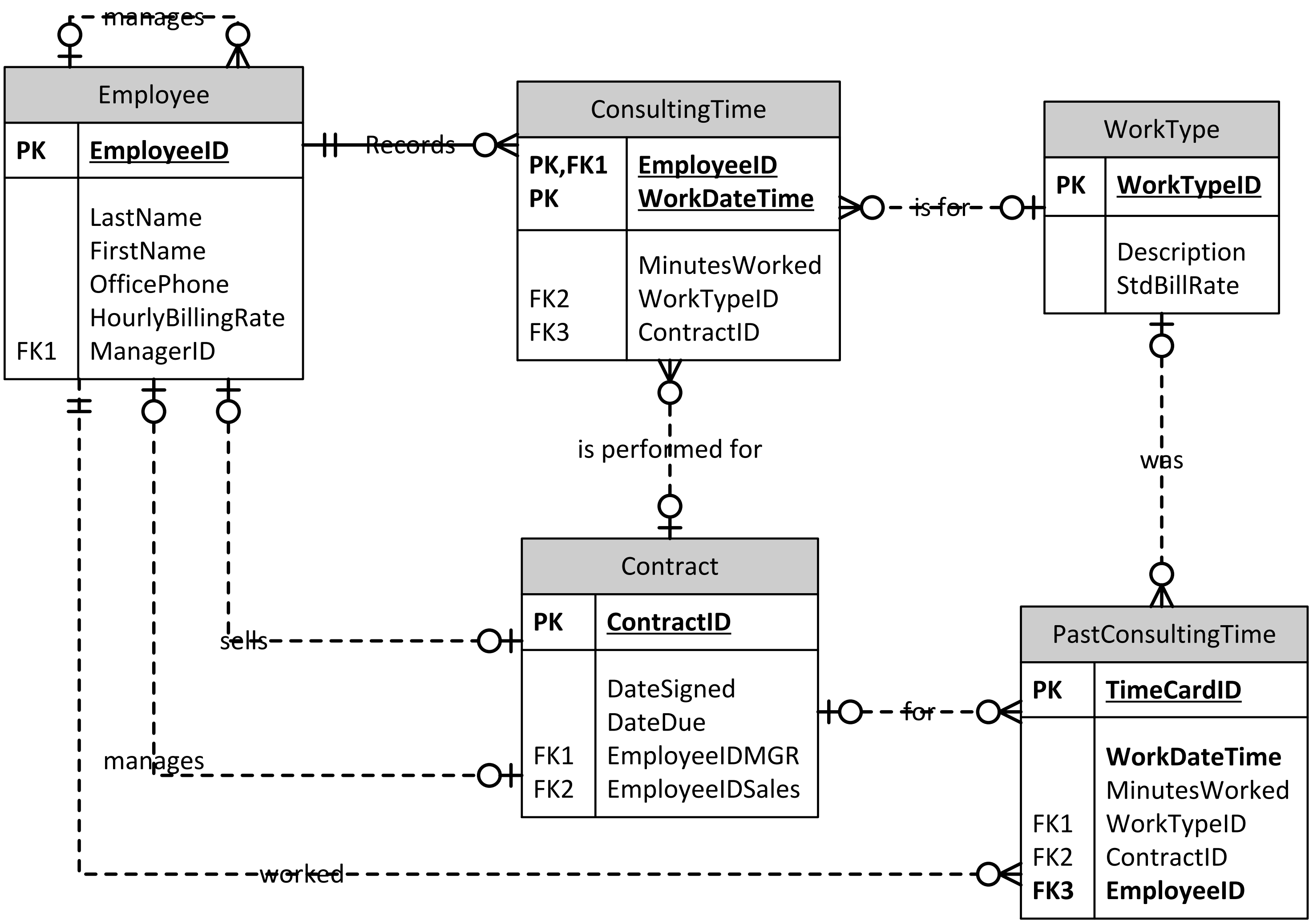
Consulting time data is associated with a given date and time (WorkDateTime).

How long should the data in that table 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



Employee	
PK	**EmployeeID**
	LastName
	FirstName
	OfficePhone
	HourlyBillingRate
FK1	ManagerID

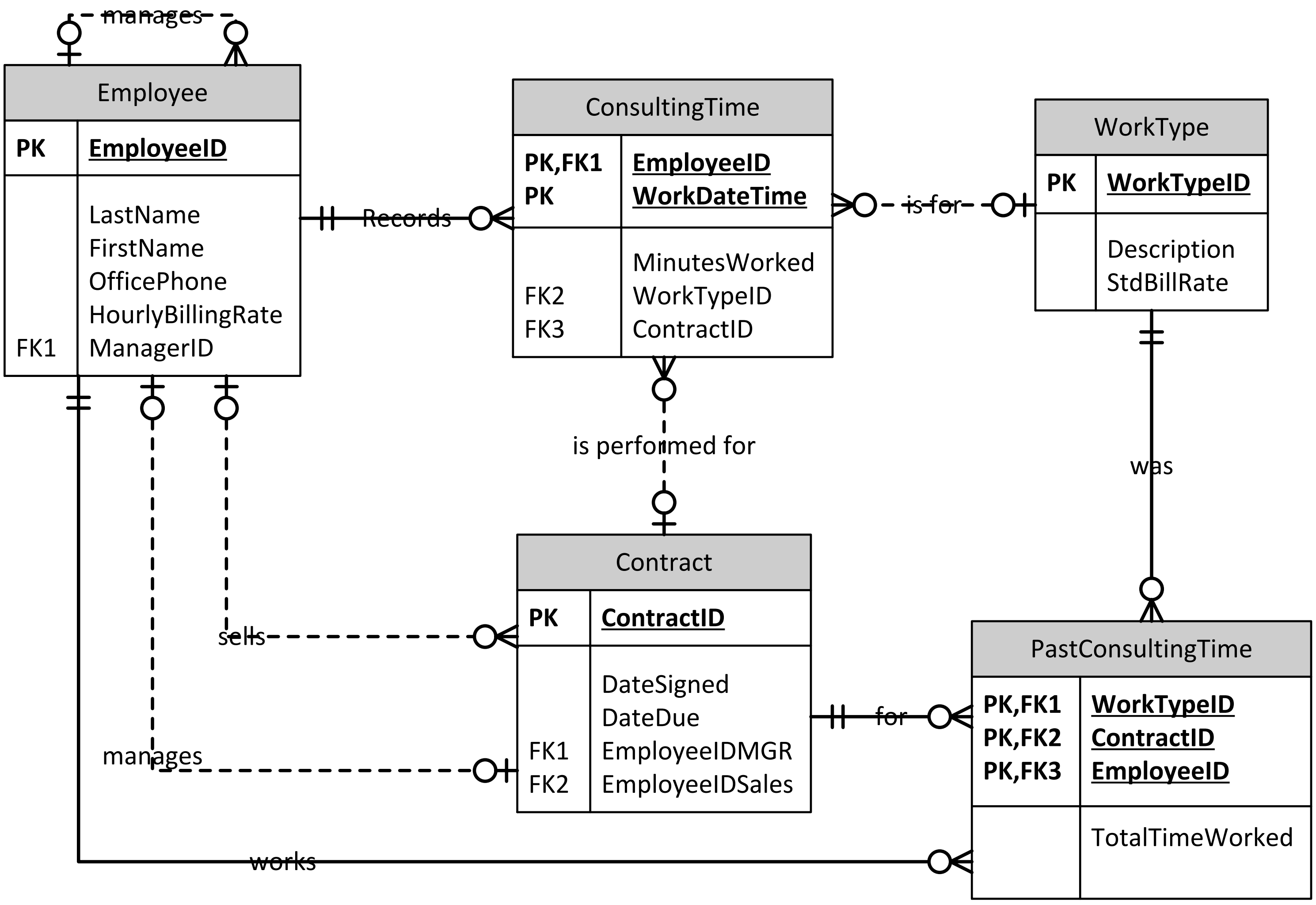
ConsultingTime	
PK,FK1	**EmployeeID**
PK	**WorkDateTime**
	MinutesWorked
FK2	WorkTypeID
FK3	ContractID

WorkType	
PK	**WorkTypeID**
	Description
	StdBillRate

Contract	
PK	**ContractID**
	DateSigned
	DateDue
FK1	EmployeeIDMGR
FK2	EmployeeIDSales

PastConsultingTime	
PK	**TimeCardID**
	WorkDateTime
	MinutesWorked
FK1	WorkTypeID
FK2	ContractID
FK3	**EmployeeID**

Summarized level of granularity by Employee, WorkType and Contract



Employee	
PK	**EmployeeID**
	LastName FirstName OfficePhone HourlyBillingRate
FK1	ManagerID

ConsultingTime	
PK,FK1	**EmployeeID**
PK	**WorkDateTime**
	MinutesWorked
FK2	WorkTypeID
FK3	ContractID

WorkType	
PK	**WorkTypeID**
	Description StdBillRate

Contract	
PK	**ContractID**
	DateSigned DateDue
FK1	EmployeeIDMGR
FK2	EmployeeIDSales

PastConsultingTime	
PK,FK1	**WorkTypeID**
PK,FK2	**ContractID**
PK,FK3	**EmployeeID**
	TotalTimeWorked

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?

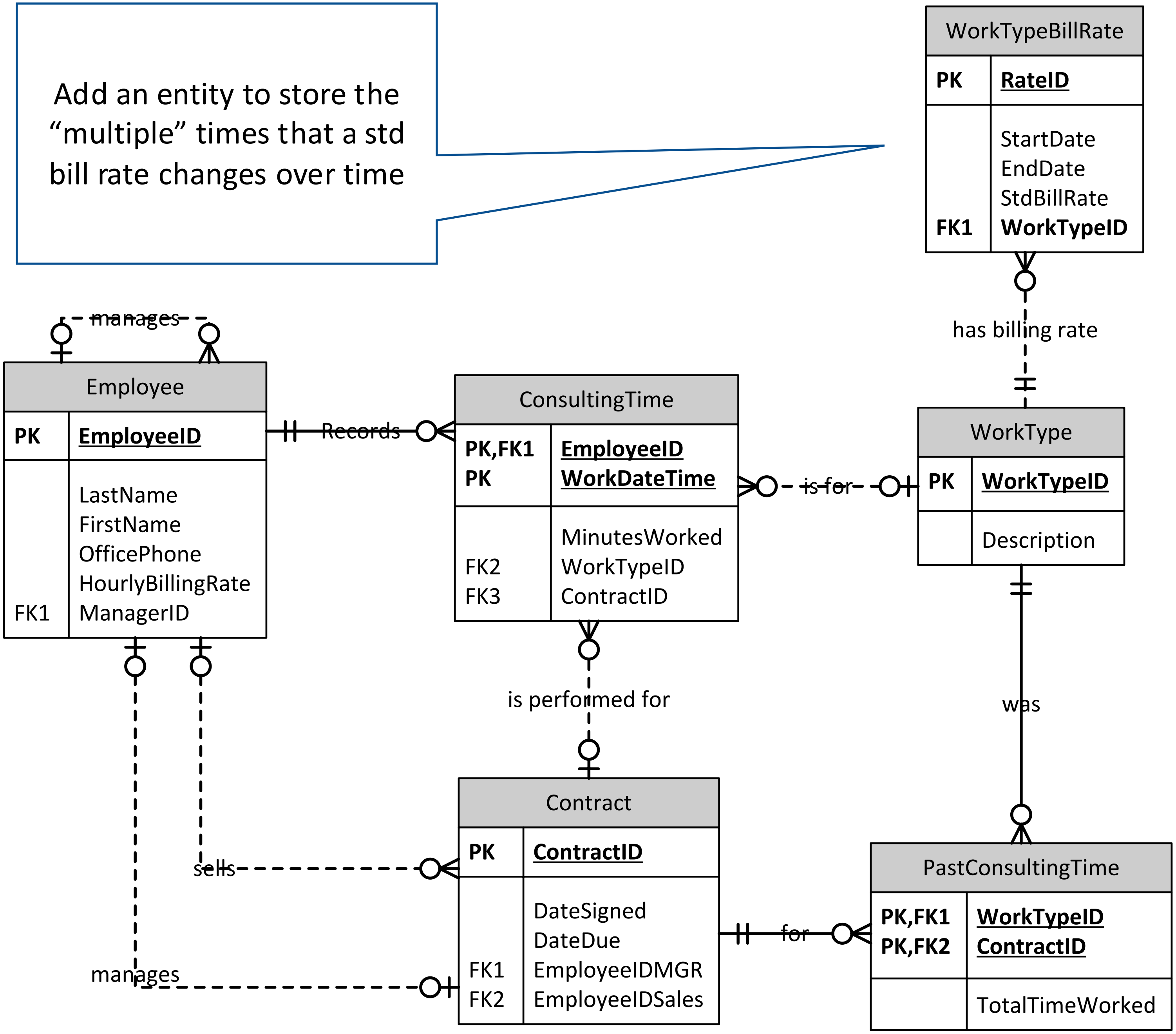
What happens when data changes over time and you want to store the changing data?

- 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**
   FK1	StartDate EndDate StdBillRate **WorkTypeID**

Employee	
PK	**EmployeeID**
    FK1	LastName FirstName OfficePhone HourlyBillingRate ManagerID

ConsultingTime	
PK,FK1 PK	**EmployeeID** **WorkDateTime**
 FK2 FK3	MinutesWorked WorkTypeID ContractID

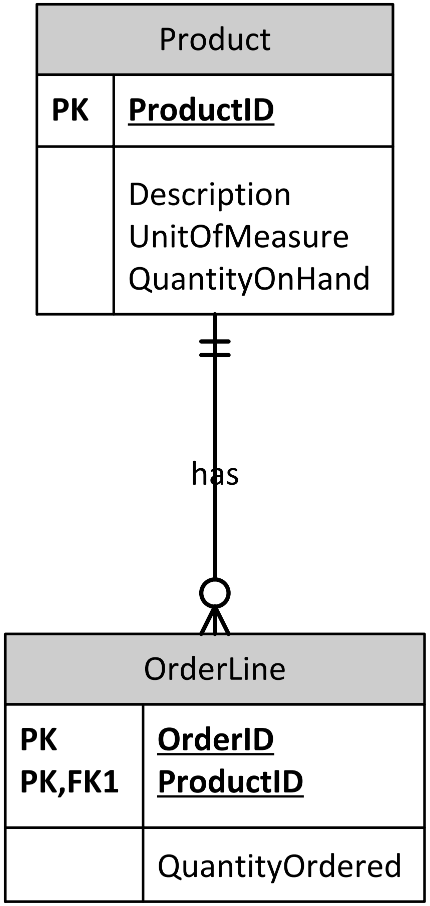
WorkType	
PK	**WorkTypeID**
	Description

Contract	
PK	**ContractID**
  FK1 FK2	DateSigned DateDue EmployeeIDMGR EmployeeIDSales

PastConsultingTime	
PK,FK1 PK,FK2	**WorkTypeID** **ContractID**
	TotalTimeWorked

Modeling Time-Dependent Data

- Status vs. event data
  - **Status** data gives a view of data at a particular point-in-time. Example: Product data that includes a current quantity on hand.
  - **Event** data is a transaction that occurs that changes the status data. For example, if an order is placed for a given product, the quantity on hand will likely decrease.
  - The database designer must decide how to store the status data, and how to store the event data.



Product	
PK	**ProductID**
	Description UnitOfMeasure QuantityOnHand

has

OrderLine	
PK **PK,FK1**	**OrderID** **ProductID**
	QuantityOrdered

Product Data: Status Data

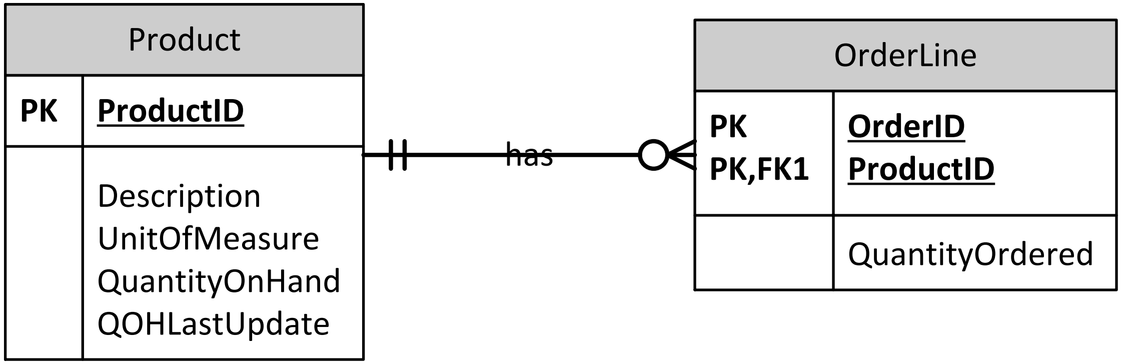
ProductID	Description	Unit of Measure	Quantity on Hand
1234-5	36” Computer Desk	Each	20
1234-8	48” Computer Desk	Each	5

Order Line Data: Event Data

OrderID	ProductID	Quantity Ordered
100	1234-5	2
100	1234-8	3

When did the event occur? Does the quantity on hand reflect the event? What are the problems involved with storing a field like “quantity on hand” with the product status data?

Event and status data must include a time stamp (a date/time field) to reconcile the status with the event.



Product

PK ProductID

Description
UnitOfMeasure
QuantityOnHand
QOHLastUpdate

has

OrderLine

PK OrderID
PK,FK1 ProductID

QuantityOrdered

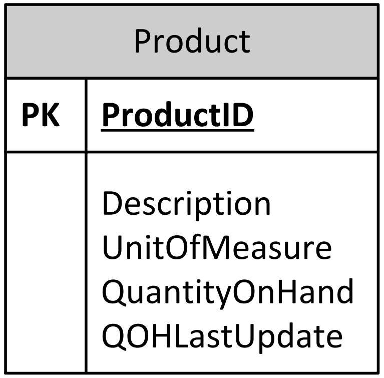
ProductID	Description	Unit of Measure	Quantity on Hand	QOH LastUpdate
1234-5	36” Computer Desk	Each	20	2/20/2024 09:00:02
1234-8	48” Computer Desk	Each	5	2/15/2024 21:15:12

OrderID	ProductID	Quantity Ordered
100	1234-5	2
100	1234-8	3

Current data that changes over time.

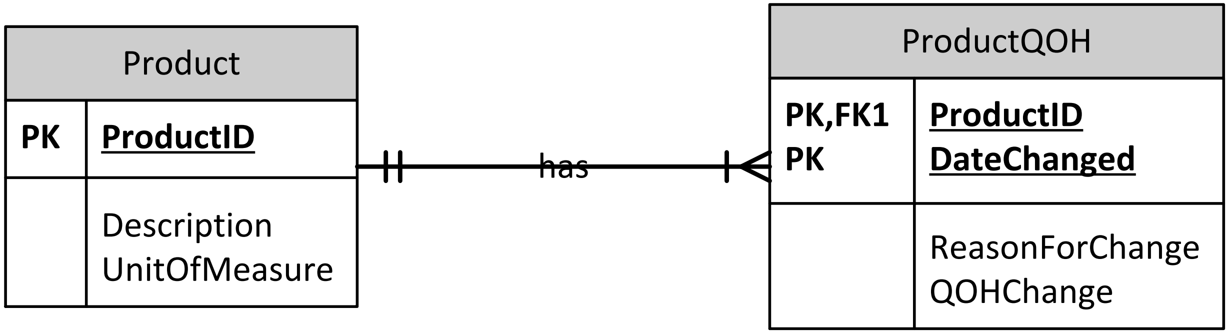
Transient vs. Periodic data: This difference is related to the processing of data.

- **Transient** data is data that will be overwritten when a change occurs to the data. For example, if we overwrite the quantity on hand, then the quantity on hand is “transient.” In SQL, this is an “update” operation.
- **Periodic** data are never changed – we maintain a history of all the changes that occur. The changes would be a multi-valued attribute so they are stored in a separate entity. For example, if we keep track of each change to the quantity on hand, then the quantity on hand is actually periodic data. In SQL, this is an “insert” operation.



Product	
PK	**ProductID**
	Description UnitOfMeasure QuantityOnHand QOHLastUpdate

Transient data with time stamp



Product	
PK	**ProductID**
	Description UnitOfMeasure

ProductQOH	
PK,FK1 PK	**ProductID** **DateChanged**
	ReasonForChange QOHChange

Periodic data with time stamp and field used to calculate the actual QOH (quantity on hand)

Design will be implemented in a relational database.

What are the characteristics of a relational database?

- All data is stored in two-dimensional tables.
- Tables are not related to each other hierarchically.
- Tables cannot inherit data from other tables.
- In most relational DBMS's there is no generalization or specialization types of relationships available.
- Tables can only be related to each other one-table-to-one-table on the basis of a foreign key.

More characteristics of a relational database

- Each column in a table must be of one primitive data type (i.e. character, integer, floating point, fixed decimal point, date, datetime, variable character)
- The quantity, data types and names of columns are established when a table is created. It is relatively difficult to add more columns.
- The quantity of rows is variable. It is very easy to add more rows.
- The intersection of a column and row (cell) can store only one value.
- The primary key (one or more columns) must contain a data value that is unique among all rows.

What makes a good design?

There is no redundancy in the rows of a single table/entity.

There is no redundancy in the rows of different tables/entities in the database.

Each of the attributes inside a given table/entity have one data value.

Each of the attributes inside a given table/entity belong together.