CS 327E Lecture 8

Shirley Cohen

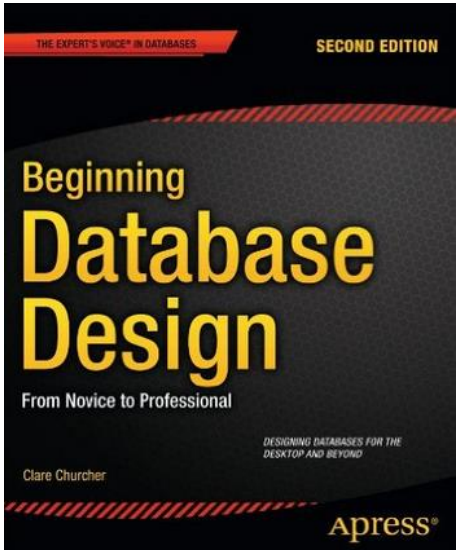
February 22, 2016

Where we are

- Phase 1: SQL
- **Phase 2: Database Design**
- Phase 3: Database-Intensive Applications

Reminders

- Homework: assigned chapters from design book
- Reading quiz at start of class
- Next midterm exam: Wednesday, March 9th



Heads-up

- Phase 2: Participation points to include in-class exercises.
- Phase 3: Build a **cool** database app in groups of 2-3.

  Start thinking of ideas now.

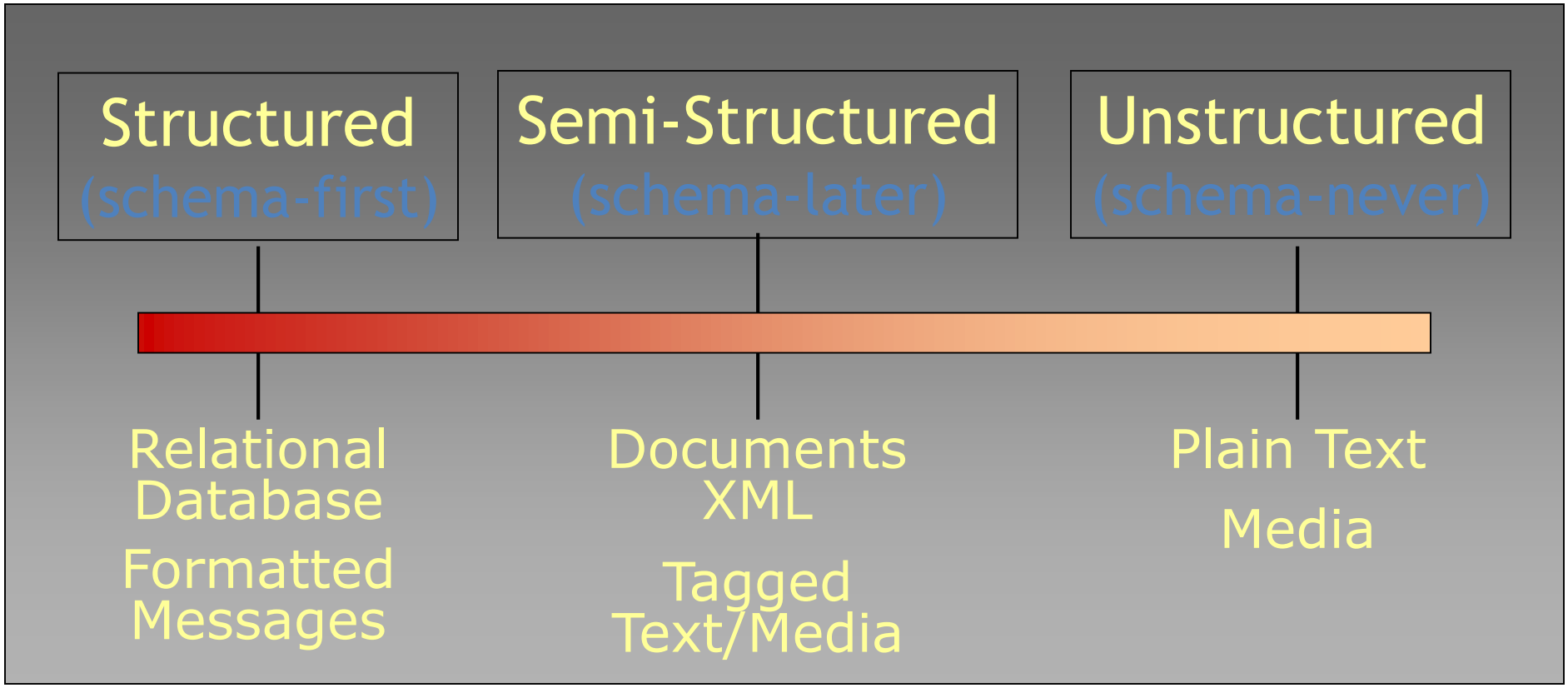
  Project guidelines will be discussed on 03/21.

Key Concepts

A *data model* is a collection of concepts for describing data.

A *schema* describes the structure of the data for a given data model.

Diversity of Data



NoSQL Systems

	Data Model
Cassandra	Columnfamily
CouchDB	Document
HBase	Columnfamily
MongoDB	Document
Neo4J	Graph
Redis	Collection
Riak	Document
Scalaris	Key/value
Tokyo Cabinet	Key/value
Voldemort	Key/value

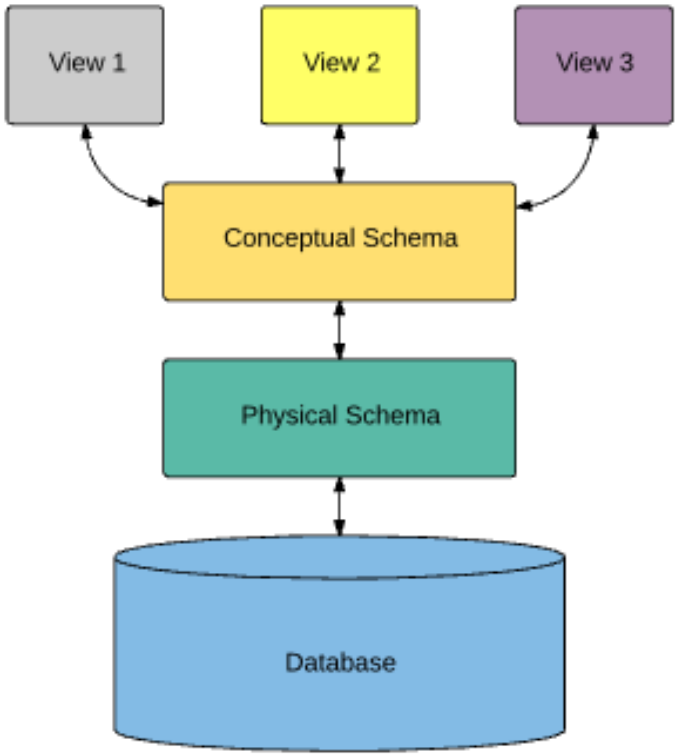
Design Process

Phase 1: Requirements Analysis

Phase 2: Conceptual Modeling

Phase 3: Physical Modeling

Phase 4: Normalization



Note: Iteration is key throughout the process, especially between phases 1 - 2 and phases 3 - 4.

Basic Constructs

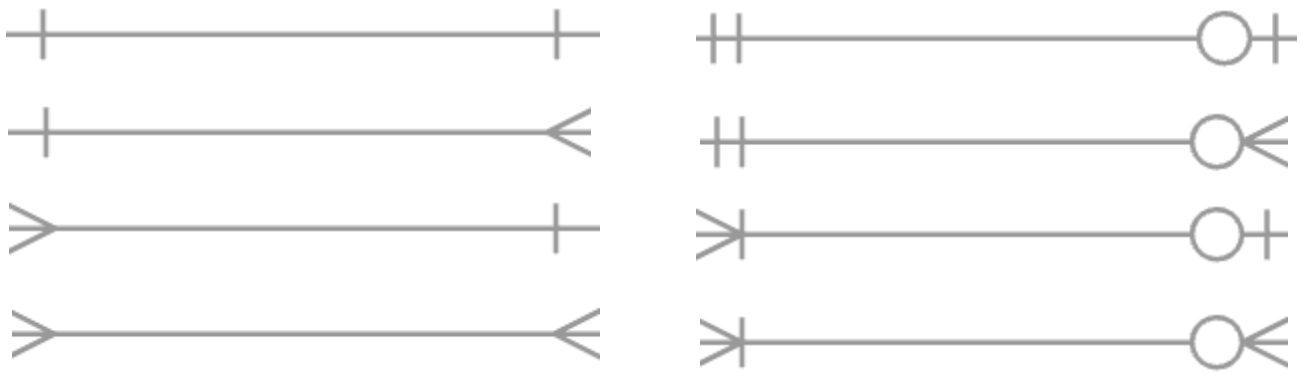
Entity = an object of interest

Attribute = property of an entity

Relationship = association between one or more entities

Relationship types:

- one-to-one:
- one-to-many:
- many-to-one:
- many-to-many:



Advanced Constructs

- Supertypes
- Subtypes

We will study hierarchies next week.

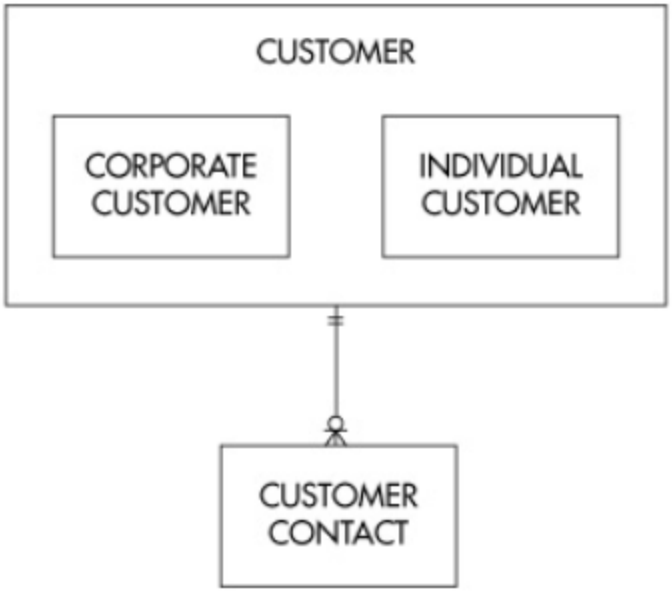
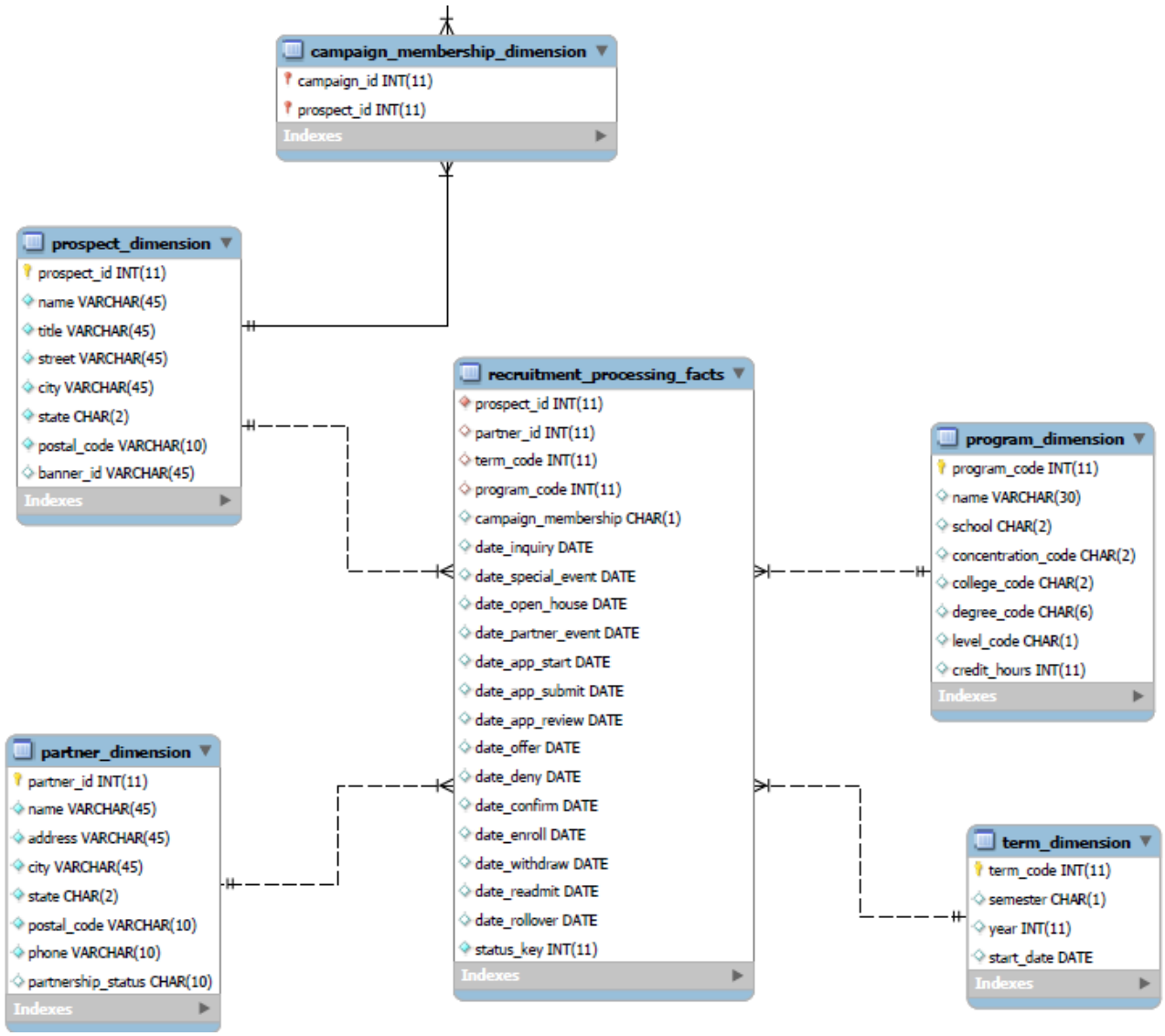


Diagram Notation

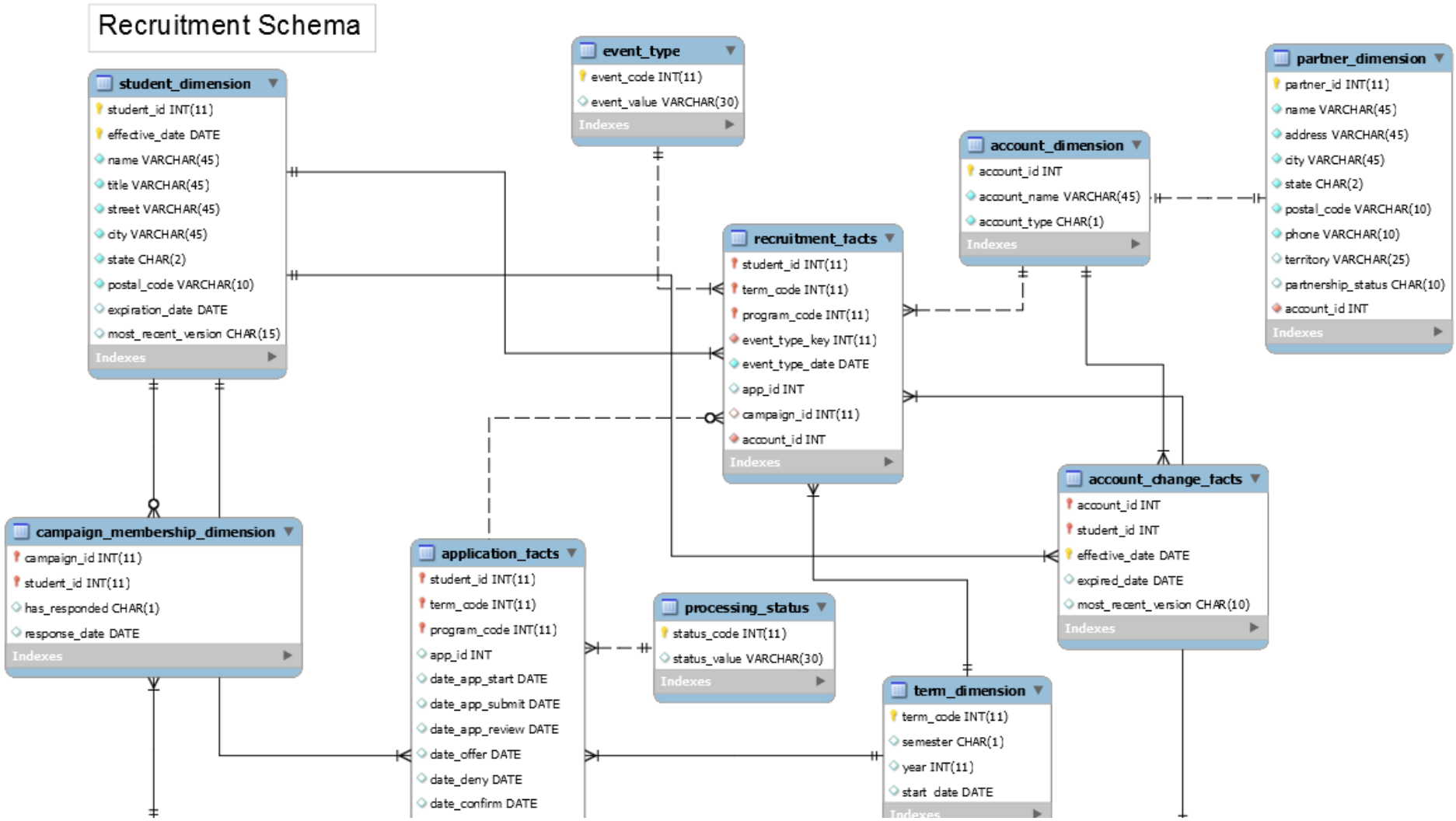
- ER diagraming tools use Chen's crow-foot notation
- UML class notation used by our book

Know how to read both notations; use only one type per diagram.

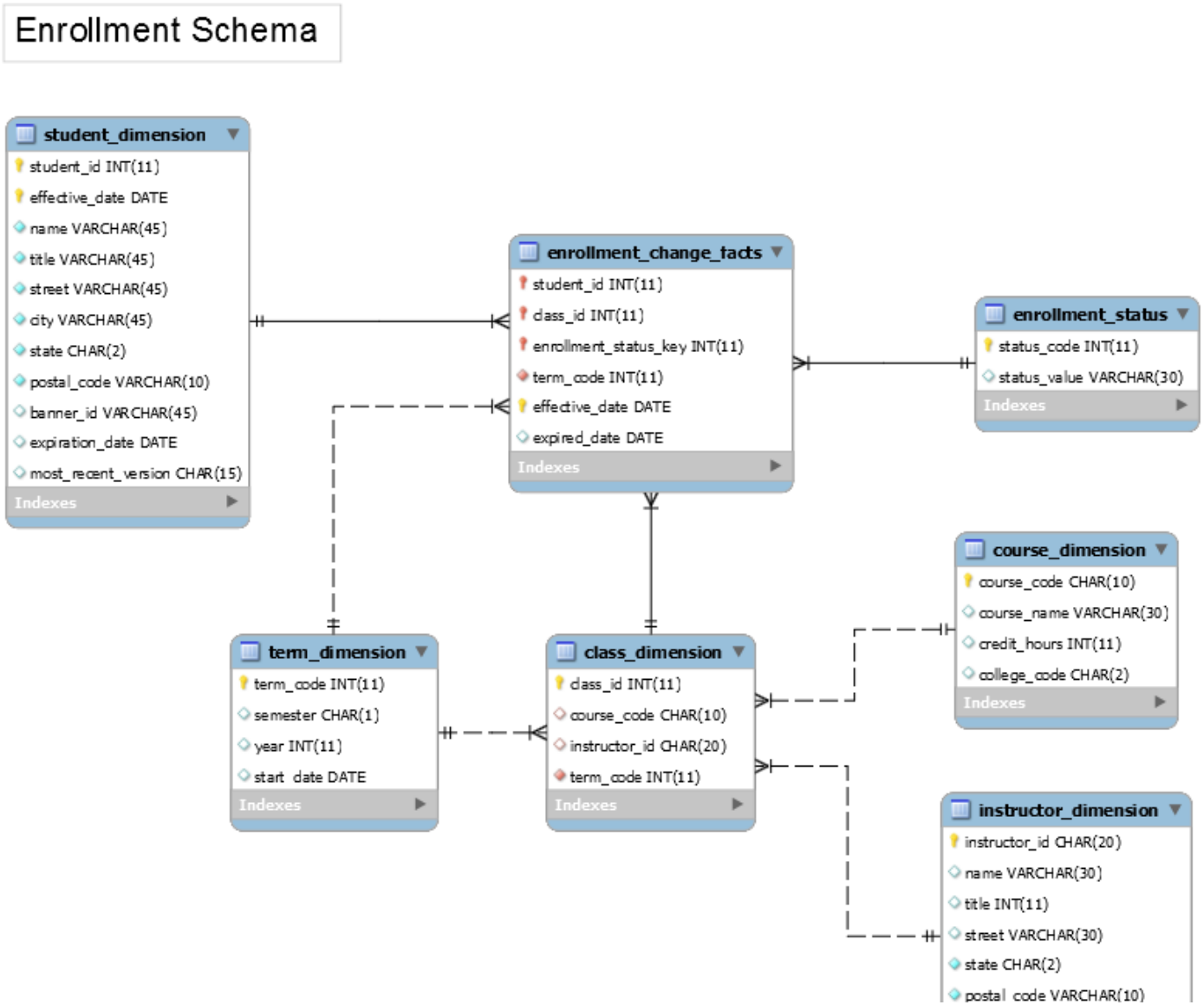
University Example: v1



University Example: v2



University Example: v2



University Example: v2

Description of Data Model:

This document contains some explanatory notes for the **Recruitment** and **Enrollment** schema diagrams.

Entities of Note:

-A *Student* in this data model is defined as anyone who expresses an interest in attending the university. It doesn't have to be someone who is enrolled in a course. We are adopting a broad definition of a student in order to reuse the same Student Dimension table across the entire lifecycle of a Student.

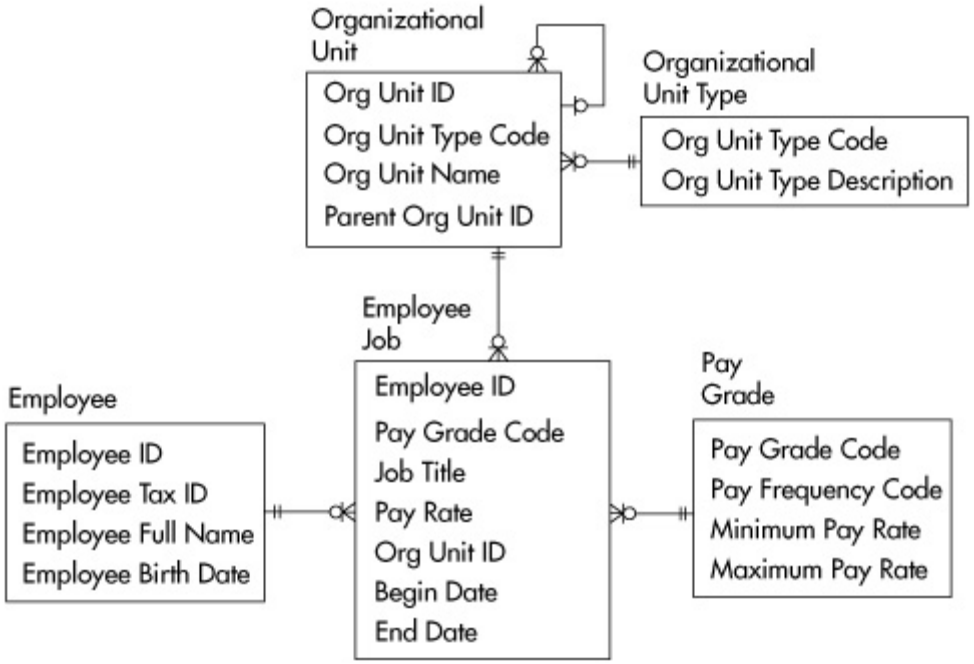
-A record in the *Recruitment Fact* table represents a single occurrence of a recruiting event. An event is defined as a event type, event date pair. The type of event is specified by the *event_type_key* field, which links to a separate lookup table *Event_Type* that contains a list of valid event types. Examples include: inquiries, open houses, partner events, etc.

-A record in the *Application Fact* table represents an instance of an application. An application cycle has several important milestones, such as app start, app submit, app confirm, app accept, etc. The presence of a date value indicates that the application has reached that particular milestone. For example, if an application has a value for *date_confirmed*, but no value for *date_enrolled*, this means that the applicant has not yet enrolled. Note that the precise enrollment history of a student is captured in the separate *Enrollment_Change_Facts*.

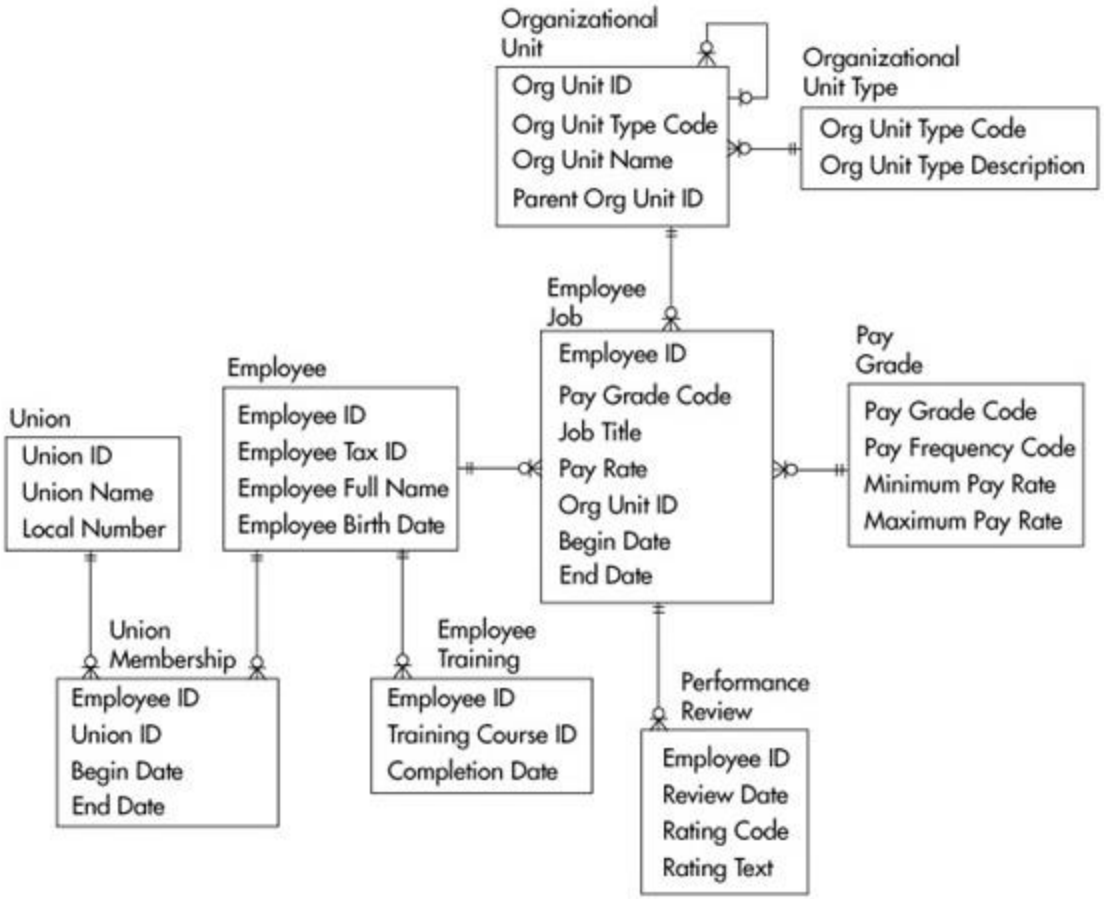
-A record in the *Enrollment_Change_Facts* table represents a class enrollment status change pertaining to a student, program, and term. Examples of a status changes include adding, dropping and withdrawing from a class.

-A *Class* is a unique instance of a *Course*. More precisely, it represents the relationship between a *course*, *instructor* and a *term*. Therefore, *students* are enrolled in *classes* and *instructors* also teach *classes*. For these reasons, an *Enrollment_Change_Fact* record links to a *Class, not a Course*.

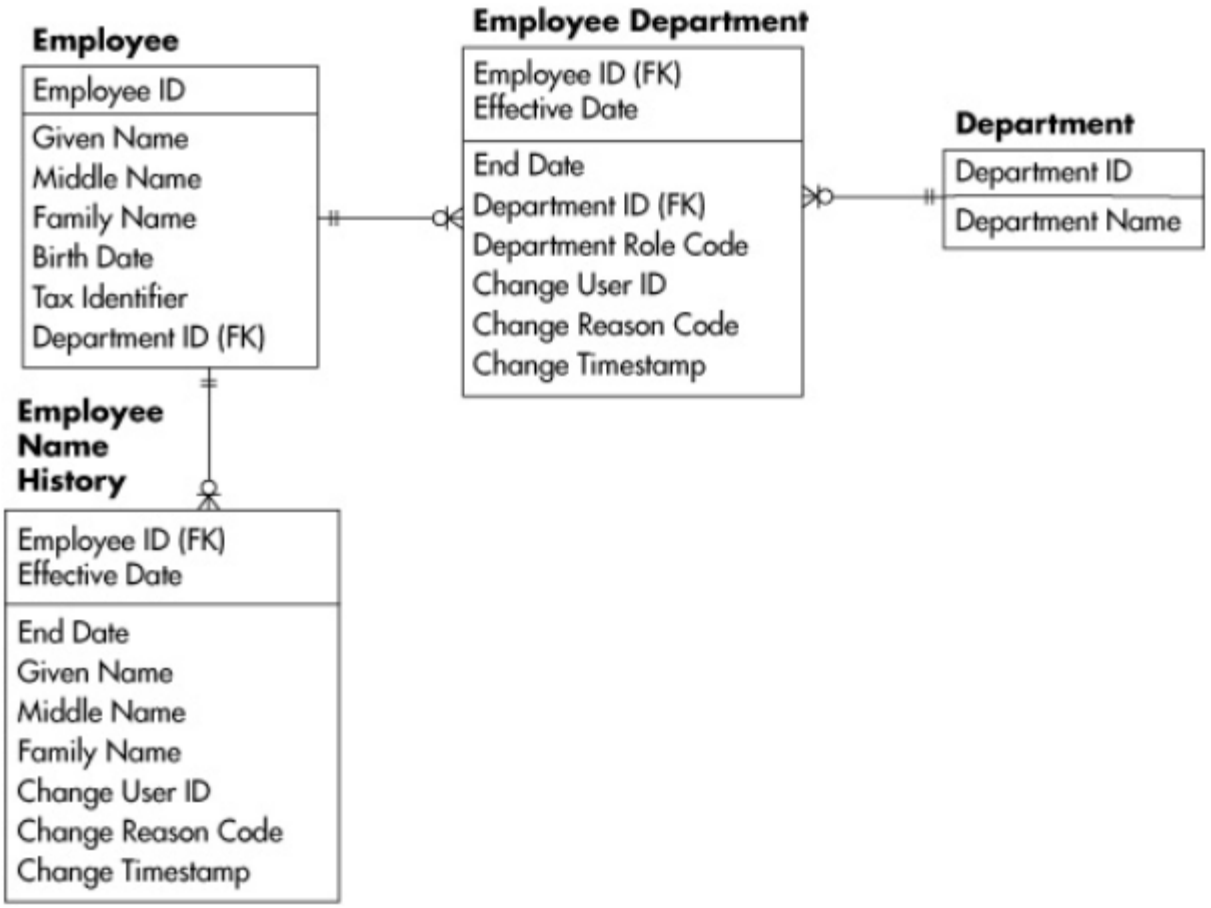
HR Example: v1



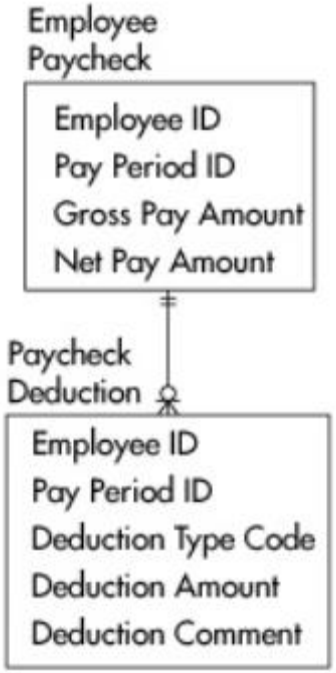
HR Example: v2



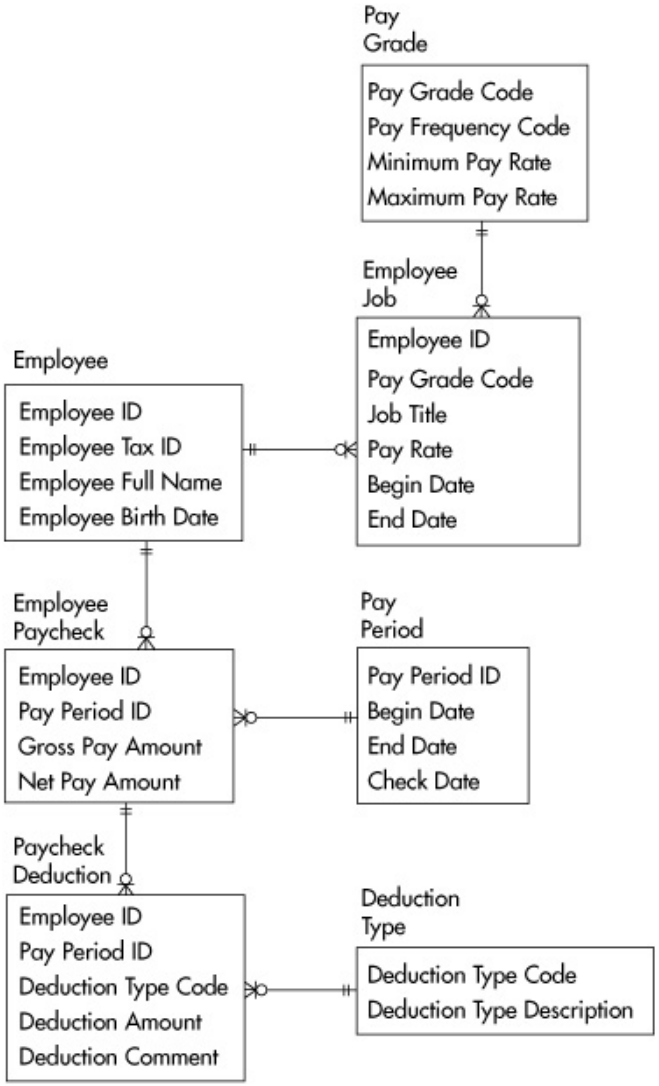
HR Example: v3



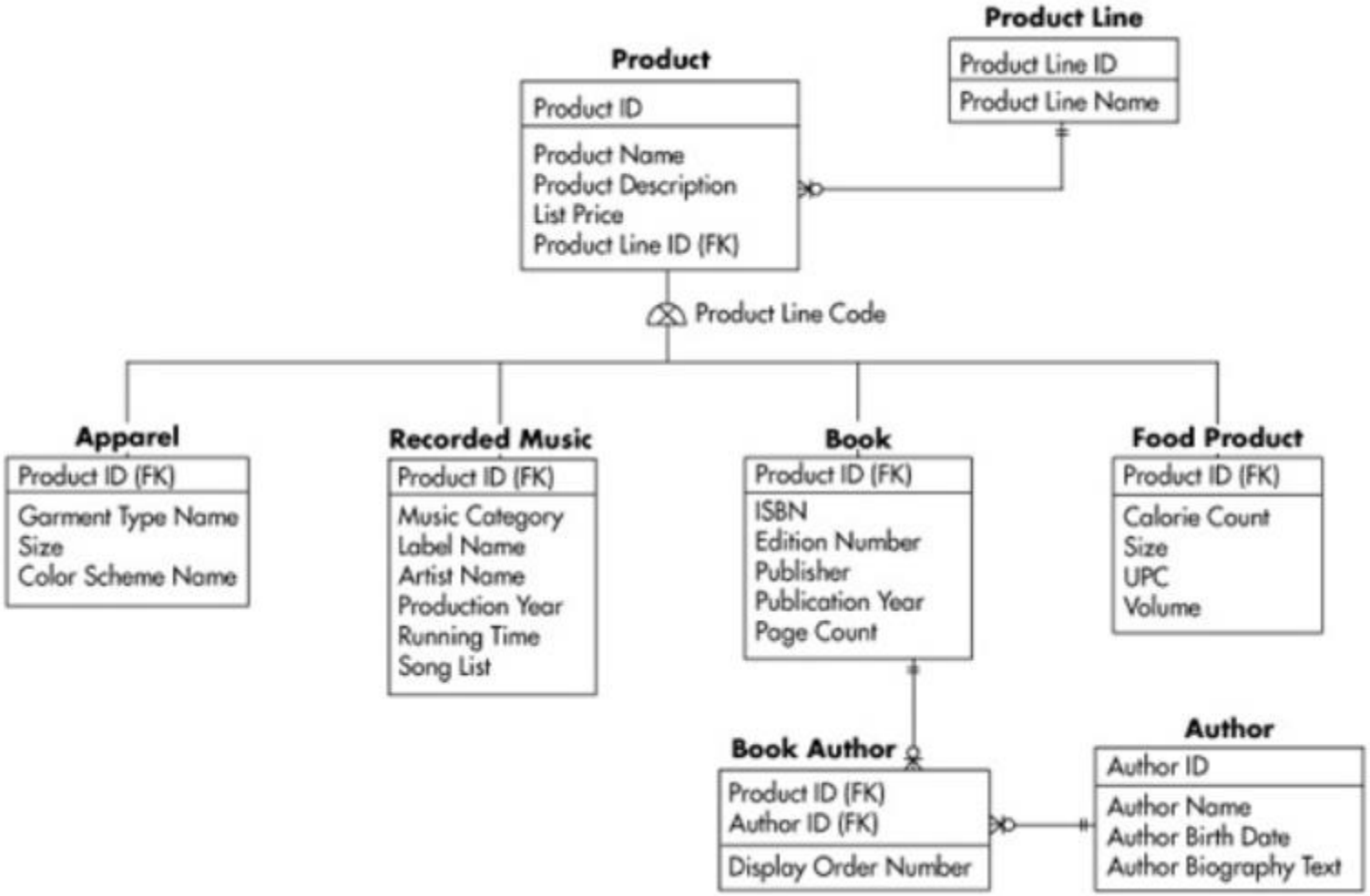
Payroll Example: v1



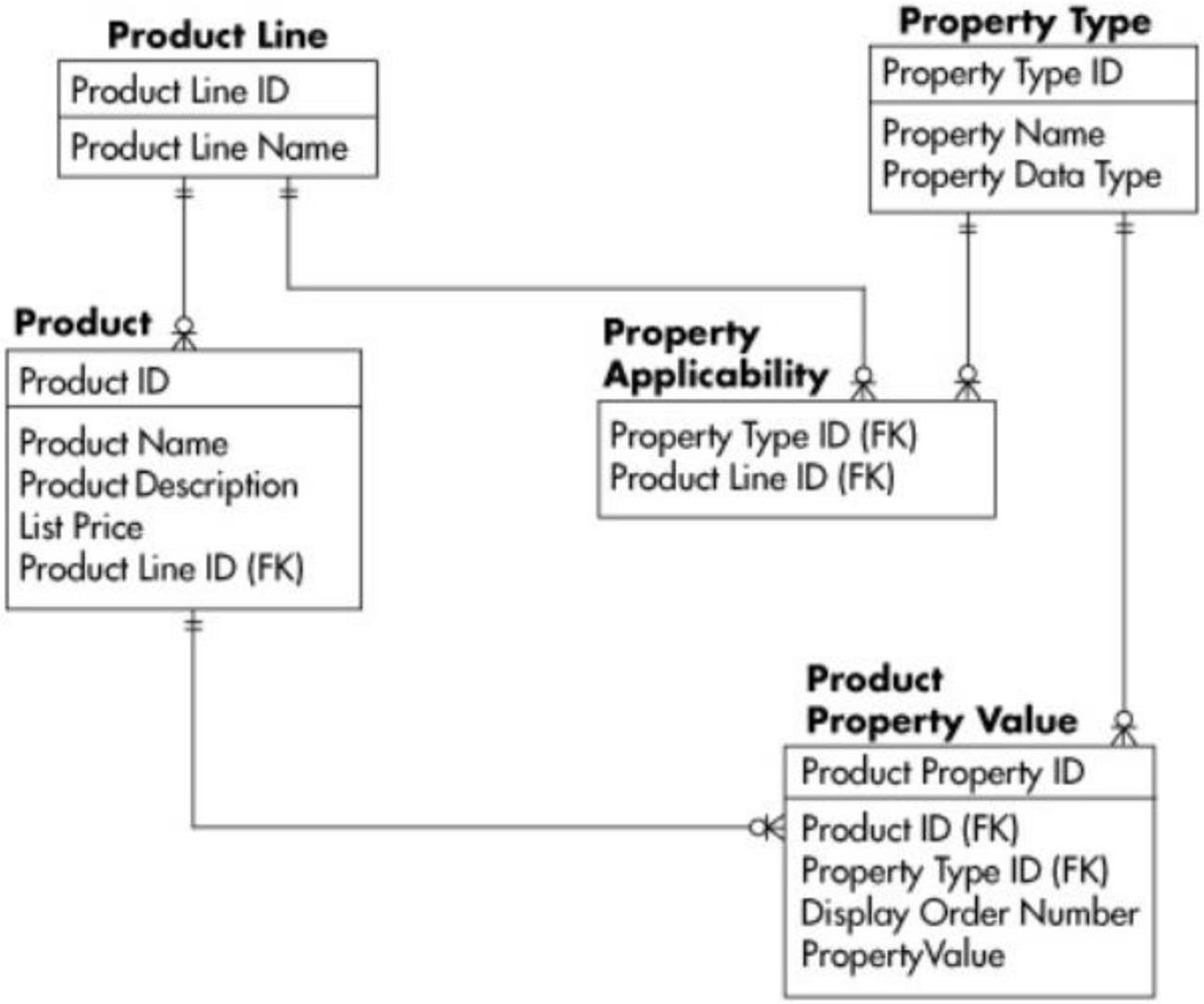
Payroll Example: v2



Product Catalog Example: v1



Product Catalog Example: v2



Product Catalog Example: v2

Product Line

Prod. Line ID	Prod. Line Desc.
101	Apparel
104	Food Products

Property Type

Prop. Type ID	Prop. Name	Prop. Data Type
P10001	Size	String
P10005	Garment Type Name	String
P10012	Calorie Count	Integer

Property Type Applicability

Prod. Line ID	Prop. Type ID
101	P10001
101	P10005
104	P10001
104	P10012

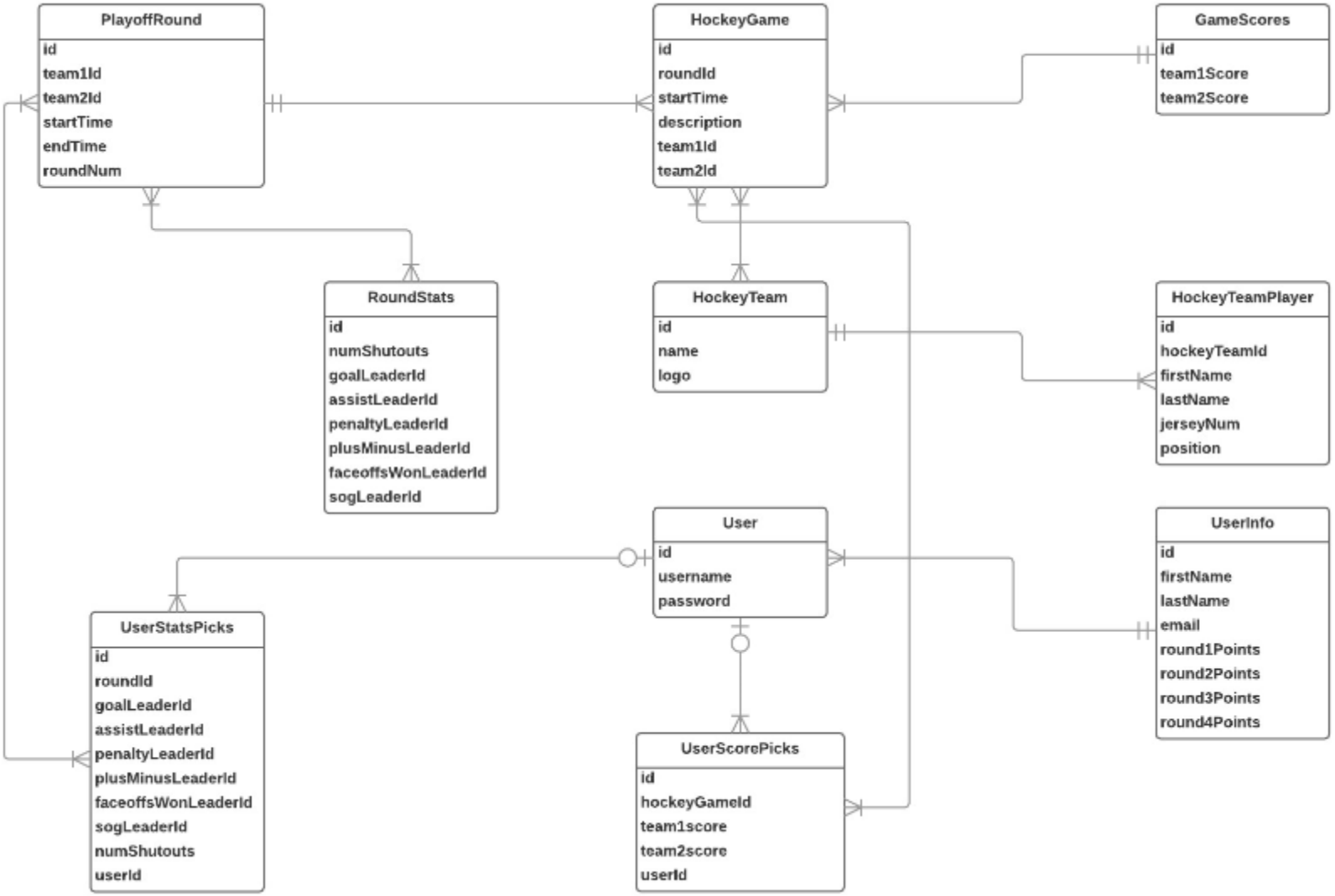
Product

Prod. ID	Prod. Name	Prod. Description	List Price	Prod. Line ID
1254678	ComfoSteer Glove	Men's leather driving glove	22.99	101
3549076	CalDry Apricots	Dried California apricots	3.25	104

Product Property Value

Prod. Prop. ID	Prod. ID	Prop. Type ID	Display Order Num.	Prop. Value
33341461	1254678	P10001	2	Large
86743573	1254678	P10005	1	Men's gloves
77303926	3549076	P10001	1	6 oz.
96901490	3549076	P10012	2	110

Hockey Example



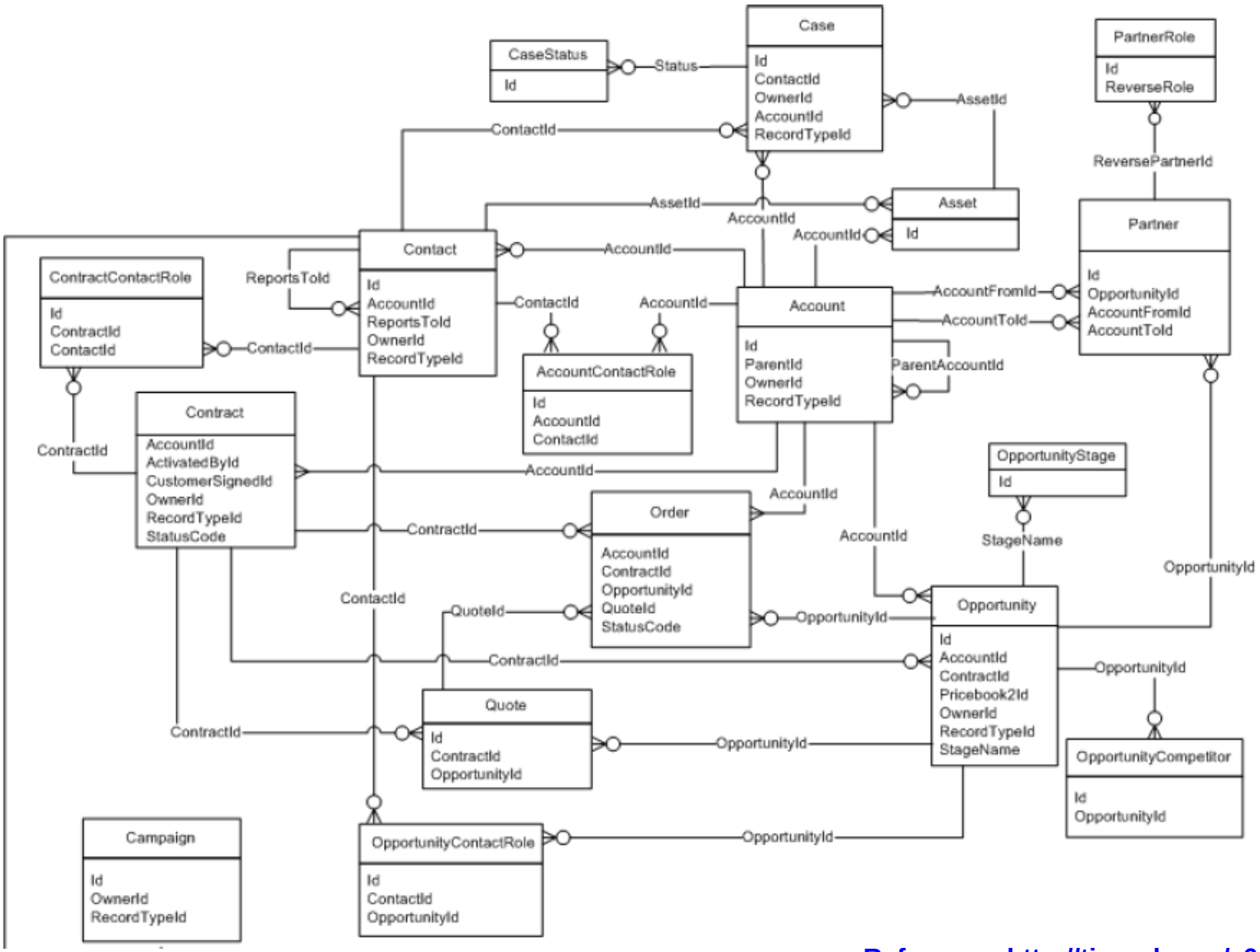
Salesforce Data Model

The entity relationship diagrams (ERDs) for standard Salesforce objects in this section illustrate important relationships between objects. The available ERDs are:

- Sales Objects—includes accounts, contacts, opportunities, leads, campaigns, and other related objects
- Task and Event Objects—includes tasks and events and their related objects
- Support Objects—includes cases and solutions and their related objects
- Salesforce Knowledge Objects—includes view and vote statistics, article versions, and other related objects
- Document, Note, and Attachment Objects—includes documents, notes, and attachments and their related objects
- User, Sharing, and Permission Objects—includes users, profiles, and roles
- Profile and Permission Objects—includes users, profiles, permission sets, and related permission objects
- Record Type Objects—includes record types and business processes and their related objects
- Product and Schedule Objects—includes opportunities, products, and schedules
- Sharing and Team Selling Objects—includes account teams, opportunity teams, and sharing objects
- Customizable Forecasting Objects—includes forecasts and related objects
- Forecasts Objects—includes objects for Collaborative Forecasts.
- Territory Management—includes territories and related objects
- Process Objects—includes approval processes and related objects
- Content Objects—includes content and libraries and their related objects
- Chatter Feed Objects—includes objects related to feeds
- Work.com Badge and Reward Objects—includes badge and reward objects
- Work.com Feedback and Performance Cycle Objects—includes feedback and performance cycle objects

Reference: http://tinyurl.com/z6t6qs4

Salesforce Sales Schema



Reference: http://tinyurl.com/z6t6qs4

Salesforce Data Dictionary

Object	Description
CampaignMemberStatus	A status value associated with a Campaign.
CampaignOwnerSharingRule	Represents the rules for sharing a Campaign with User records other than the owner.
CampaignShare	Represents a list of access levels to a Campaign along with an explanation of the access level. For example, if you have access to a record because you own it, the Access Level value is `Full` and Reason for Access value is `Owner`.
CampaignTag	Associates a word or short phrase with a Campaign.
Case	A customer issue such as a customer's feedback, problem, or question.
CaseArticle	Represents the association between a Case and a KnowledgeArticle. This object is available in API version 20.0 and later.
CaseComment	A comment that provides additional information about the associated Case.
CaseContactRole	The role that a given Contact plays on a Case.
CaseFeed	Represents a single feed item in the feed displayed on the detail page for a case record. This object is available in API version 18.0 and later.
CaseHistory	Historical information about changes that have been made to the associated Case.
CaseMilestone	Represents a milestone (required step in a customer support process) on a Case. This object is available in API version 18.0 and later.
CaseOwnerSharingRule	A rule that grants access to a case to users other than the owner.

Reference: http://tinyurl.com/z6t6qs4

Design Tips

- **Tip 1.** Clearly state the database requirements: what data and updates go into the database and what data and queries come out of the database.
- **Tip 2.** Best order of modeling: 1-entities, 2-relationships, 3-attributes and 4-user views.
- **Tip 3.** Keep ER diagram to one page. Accompany diagram with descriptions, assumptions and explanation in supporting a document.
- **Tip 4.** Interact frequently with end-users.

In-Class Exercise

Goal: Design a schema for a *Personal Time Assistant.*

A *Personal Time Assistant* is a database that solves the problem of time-management by helping us track our time commitments and making sure that they happen.

High-level system requirements:

1. Represents every type of commitment that competes for your time.
2. Represents temporal scope of a commitment. For example: “I intend to ride my bike for 2-3 hours sometime this weekend”.
3. Schedules tasks (and other commitment types) and monitors their progress until they get done.
4. Corrects common mistakes (e.g. procrastination and over-estimating future availability).

Homework for Next Time

- Read chapters 4 and 5 from the Beginning Database Design book
- Exercises at the end of chapters

Resources & References

- Lots of common entity types (e.g. Customer, Product, Event, etc.): http://schema.org
- ER diagramming tools: MySQL Workbench and LucidChart
- Survey paper: J. Hellerstein and M. Stonebraker. "What Goes Around Comes Around" in Readings in Database Systems, 2004.
- Supplemental book: Andy Oppel's Data Modeling A Beginner's Guide (2009).