Database Design and Administration for OnBase WorkView Solutions

Mike Martel – Senior Project Manager

Agenda

1. Solution Design vs. Database Design
2. Data Modeling/Design Concepts
3. ERD Diagramming Labs
4. WorkView Class Design
5. WorkView Filter Design
6. Database Administration

Solution Design

Conceptualizing the Solution

“

Measure twice and cut once.

”

- Unknown

English Proverb

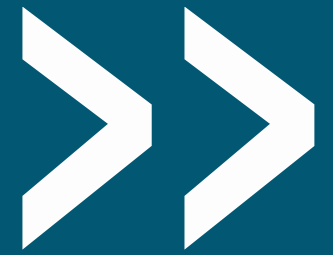
Solution Design

- Refers to the process of mapping the business requirements to specific OnBase technologies, and defining the various system components that will need to be configured in OnBase (I.E. Doc Types, Applications, Reports, etc.)

Database Design

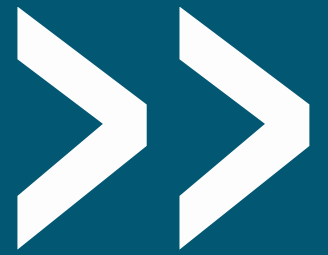
Defining the data requirements

https://www.youtube.com/watch?v=RNJl9EEcsoE



Data Modeling/Design Concepts

Defining the data requirements



The Relational Model

METHOD OF REPRESENTING DATA CONCEPTUALLY AS TABLES (Relations) WITH SPECIFIC PROPERTIES (Attributes and Data Types).

DATA MANIPULATION IS DEFINED IN TERMS OF THE MANIPULATION OF THESE TABLES OF DATA.

RELATION = TWO-DIMENSIONAL TABLE

RELATION IS COMPOSED OF COLUMN HEADINGS (FIELDS, ATTRIBUTES) WITH ROWS OF DATA (RECORDS, TUPLES)

Table = Relation

ORDER

Column = Field = Attribute

OrderId	OrderDate	CustomerId
001	010188	1111
002	020389	2222
003	030389	1111

Row = Record = Tuple

Entries within a column have the same data type. More specifically they are defined on the same **Domain** or pool of values

The Relational Model

TABLE PROPERTIES:

- EACH CELL CONTAINS A SINGLE VALUE
- ENTRIES IN A COLUMN ARE OF THE SAME DOMAIN
- EACH COLUMN HAS A UNIQUE NAME
- ORDER OF THE COLUMNS AND ROWS HAS NO SIGNIFICANCE
- NO TWO ROWS ARE IDENTICAL
- TABLE HAS UNIQUE PRIMARY KEY

ILLEGAL RELATION

ID	NAME	COURSE
1111	JOE	MIS380, MIS225
2222	SAM	MIS202, MIS420
3333	SALLY	MIS380

Cannot have multiple values in a single cell

LEGAL RELATION

COURSE	TITLE	HOURS
MIS380	Database I	4
MIS225	Visual Basic	4
MIS420	Systems II	4

One value in each cell –
this is called First Normal Form (1NF)
when a table has no repeating values in a cell

Primary Keys

- **Primary Key = One or more attributes that uniquely identify each row in the table.**
- **Primary keys have four properties:**
  - Unique
  - Non-null (not blank)
  - Minimal - use the minimum number of attributes necessary for uniqueness
  - Non-updatable - the value of the primary key should never change - avoid things like names and descriptions that might change over time

Database Design

- Databases are designed as a group of related tables.
- Entities are like nouns—they are the real-world persons, places or things being modeled in the database—e.g., employees, orders, books
    - Each table is a collection of all the data (attributes) to be stored about a particular entity
- Relationships are like verbs—they show action or possession—that related one entity to another (e.g., a customer places orders; an employee has dependents; a doctor treats patients and a patient can visit multiple doctors)
    - Relationships can be one to one, one to many, or many to many
- The process of database design involves the grouping of data items into a single set of entities(tables) and implementing the relationships between those entities by duplicating the primary keys from one table into another

Entity-relationship model (E-R model)

A logical representation of the data for an organization or for a business area using entities for categories of data and relationships for association between entities.

Entity-relationship diagram (ERD)

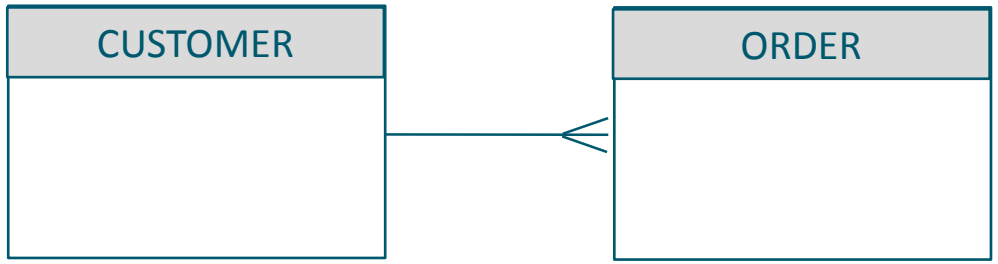
A graphical representation of an entity-relationship model.

Entity-Relationship Diagrams

- A visual model of the entities and their relationships is called an Entity Relationship Diagram (ERD)
- Each entity is diagramed as a box with the name of the entity at the top in capital letters (these will ultimately become the names of tables)
- The attributes of the entity (data items that will be collected and stored about the entity) are listed in the box. Each attribute must be uniquely named (these will become columns in the table)
- The primary key is bold and underlined.

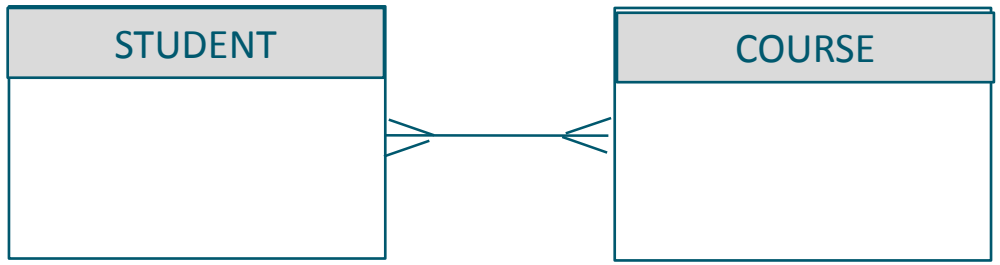
Entity-Relationship Diagrams

- Relationships are identified by their cardinality
  - One to many – a customer places many orders and a particular order is placed by one customer



Crow's Foot represents the many side of the relationship

  - Many to many – a student takes many courses and a course is taken by many students



One to Many Relationships

- Suppose we need to represent that customer 1 has placed orders 100 and 300 and customer 2 has placed order 300.
- So far we have the two tables with a primary key identified for each table

CUSTOMER

CustomerId	FirstName	LastName	Phone
1	John	Day	592-0646
2	Thom	Luce	592-1111
3	Sean	McGann	592-2222

ORDER

OrderId	Date	Total
100	10/29/08	50
200	10/30/08	65
300	11/2/08	45

- How do we know which order belongs to which customer by looking at these tables?

One to Many Relationships

- A one to many relationship can be created by duplicating the primary key from one table into the other

CUSTOMER

CustomerId	FirstName	LastName	Phone
1	John	Day	592-0646
2	Thom	Luce	592-1111
3	Sean	McGann	592-2222

ORDER

OrderId	Date	Total	CUSTOMER$CustomerID
100	10/29/08	50	2
200	10/30/08	65	3
300	11/2/08	45	2

- Here we have added the CUSTOMER$CustomerId column to the ORDER table and in it we have placed the primary key of the customer placing the order.
- Now we can tell who made each order

One to Many Relationships

- This illustrates the critical role primary keys play in the relational model – the duplication of primary key values is how relationships get created.
- A field in a table that is a duplicate of a primary key is called a foreign key.
- Foreign keys are named TABLE$primarykey to make them easier to spot.

CUSTOMER

CustomerId	FirstName	LastName	Phone
1	John	Day	592-0646
2	Thom	Luce	592-1111
3	Sean	McGann	592-2222

ORDER

OrderId	Date	Total	CUSTOMER$CustomerID
100	10/29/08	50	2
200	10/30/08	65	3
300	11/2/08	45	2

One to Many Relationships

- What if we tried to duplicate the primary key from the Order table into the Customer table to create the relationship?

CUSTOMER

CustomerId	FirstName	LastName	Phone	ORDER$OrderId
1	John	Day	592-0646	
2	Thom	Luce	592-1111	100,300
3	Sean	McGann	592-2222	200

ORDER

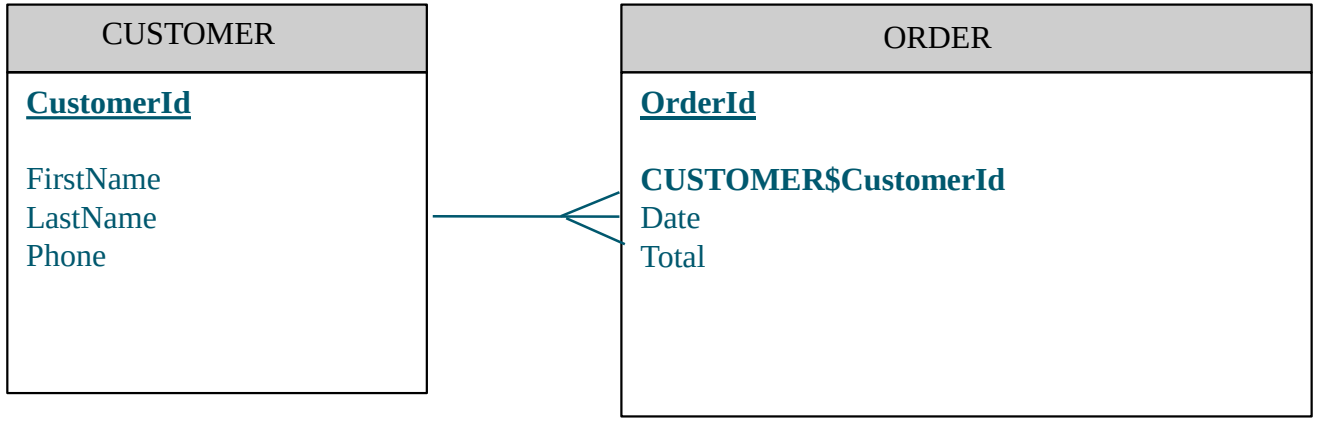
OrderId	Date	Total
100	10/29/08	50
200	10/30/08	65
300	11/2/08	45

Note how this would violate the requirement that a cell in the table can only contain a single value.

- There is only one way to implement a one to many relationship:
  - The key from the table on the one side of the relationship is duplicated into the table on the many side as a foreign key

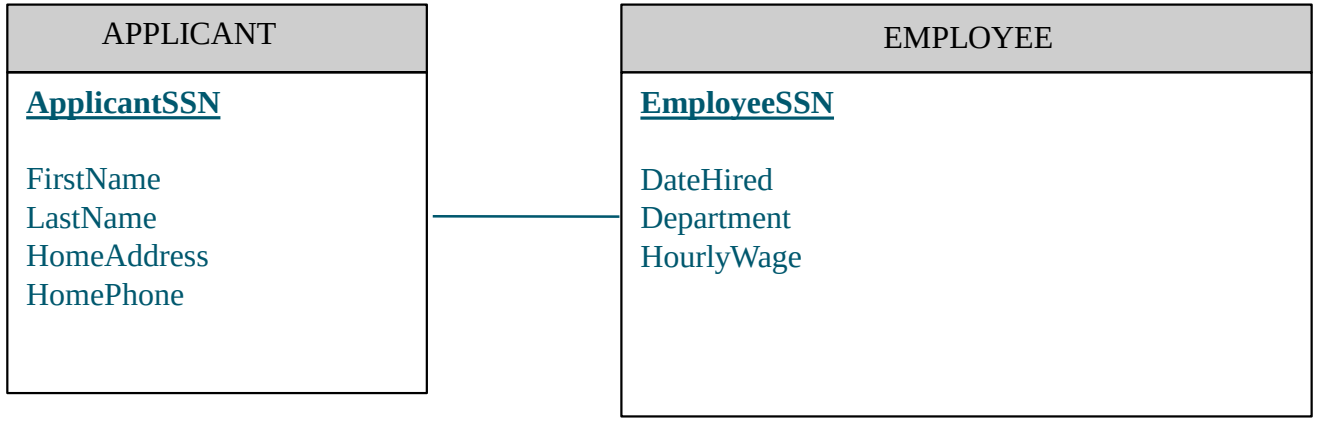
One to Many Relationships

- Other ways to remember how to implement a one to many relationship
  - The direction of the duplication of the primary key is the same direction the crow's foot is pointing
  - The duplication is also always from the parent to the child



One to One Relationships

- Another Relationship Type is the one to one relationship
  - Every record in one table has one, and only one, record in the second table
  - This is considered to not be a good design and is usually resolved by combining both tables into one single table.



Many to Many Relationships

- Suppose we have students and courses and need to represent that student 11111 has completed MIS380 with an A and MIS400 with a B and that student 22222 has completed MIS380 with a B+.

STUDENT

StudentId	FirstName	LastName
11111	John	Day
22222	Thom	Luce
33333	Sean	McGann

COURSE

CourseCode	Title	Hours
MIS380	Database	4
MIS400	Applications	4
MIS420	Systems	4

- Let’s try the same approach we used with the one to many relationship

Many to Many Relationships

- Duplicating the CourseCode into the Student table runs into the problem of violating the restriction of one value in each table cell

STUDENT

StudentId	FirstName	LastName	CourseCode
11111	John	Day	MIS380, MIS400
22222	Thom	Luce	MIS380
33333	Sean	McGann	

COURSE

CourseCode	Title	Hours
MIS380	Database	4
MIS400	Applications	4
MIS420	Systems	4

Many to Many Relationships

- Duplicating the CourseCode into the Student table runs into the problem of violating the restriction of one value in each table cell since students are going to need to be associated with many courses

STUDENT

StudentId	FirstName	LastName	CourseCode
11111	John	Day	MIS380, MIS400
22222	Thom	Luce	MIS380
33333	Sean	McGann	

COURSE

CourseCode	Title	Hours
MIS380	Database	4
MIS400	Applications	4
MIS420	Systems	4

Many to Many Relationships

- Duplicating the StudentId into the Course table runs into the same problem since many students are going to need to be associated with a course

STUDENT

StudentId	FirstName	LastName
11111	John	Day
22222	Thom	Luce
33333	Sean	McGann

COURSE

CourseCode	Title	Hours	StudentId
MIS380	Database	4	11111,22222
MIS400	Applications	4	11111
MIS420	Systems	4	

Many to Many Relationships

- The solution is to create a third table that contains a duplicate of the primary key from both of the tables in the many to many relationship – these are foreign keys.
- This new table is called an Associative table
- Note that this also creates a table where the grade can be recorded - you need both a student and a course to be able to determine a grade

STUDENT

StudentId	FirstName	LastName
11111	John	Day
22222	Thom	Luce
33333	Sean	McGann

COURSE

CourseCode	Title	Hours
MIS380	Database	4
MIS400	Applications	4
MIS420	Systems	4

GRADE

COURSE$CourseCode	STUDENT$StudentId	Grade
MIS380	11111	A
MIS400	11111	B
MIS380	22222	B+

Many to Many Relationships

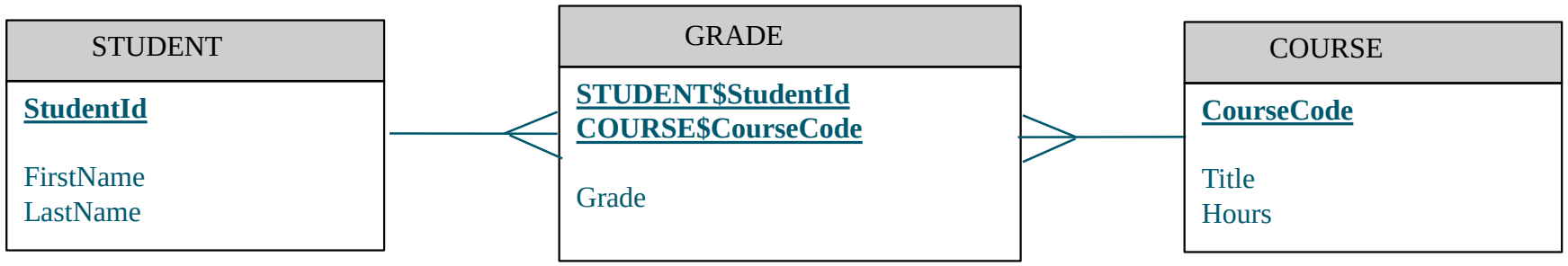
GRADE

COURSE$CourseCode	STUDENT$StudentId	Grade
MIS380	11111	A
MIS400	11111	B
MIS380	22222	B+

- The primary key of this table is the combination of the CourseCode and the StudentId.
  - The course code by itself is not unique - the course repeats for every student that takes the course
  - The student id by itself is not unique – a student will appear each time they take a course
  - But the combination of course code and student id is unique
  - This is known as a concatenated key
  - Both halves of the concatenated key are also foreign keys

Many to Many Relationships

- The associative table basically replaces the many to many relationship with two one to many relationships

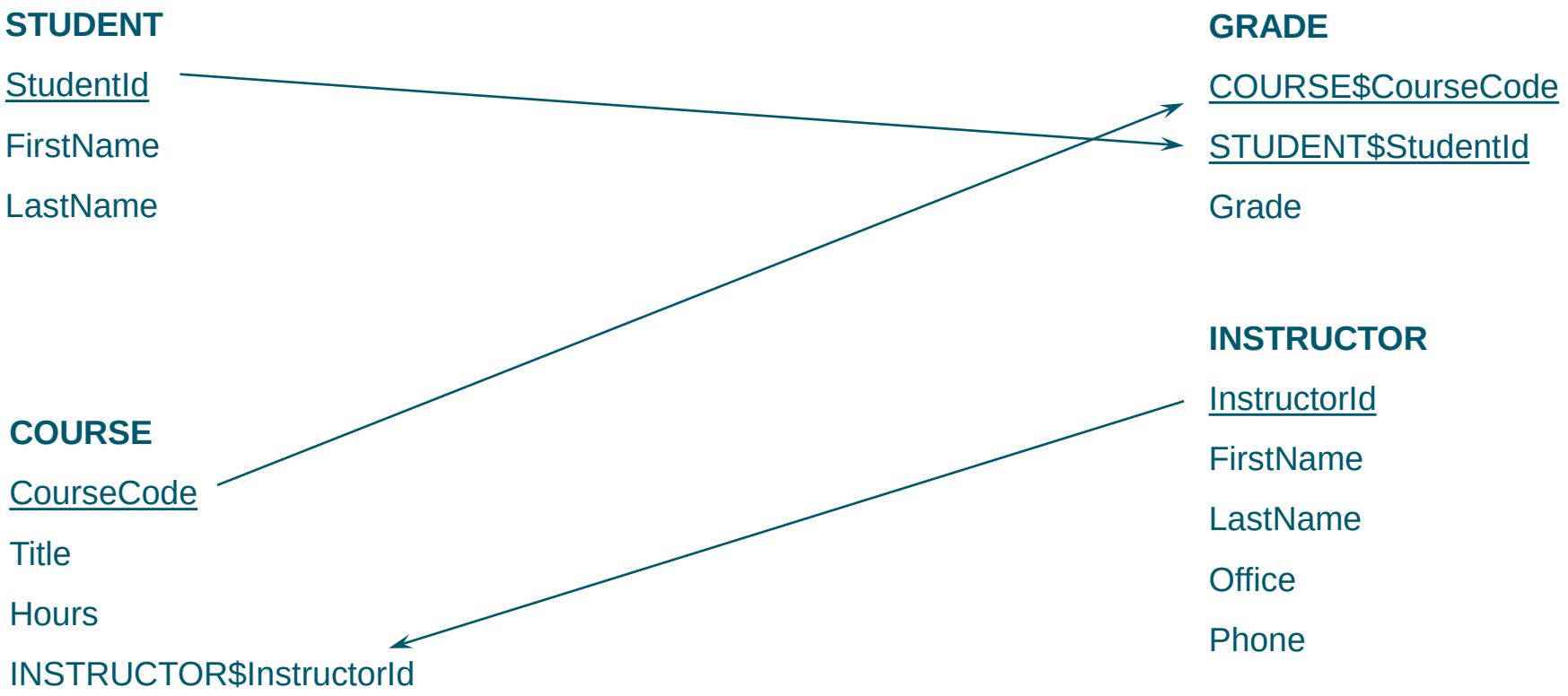


- A student can be matched to many rows in this new Grades table and a row in the Grades table matches one student
- A course can be matched to many rows in this new Grades table and a row in the Grades table matches one course

GRADE

COURSE$CourseCode	STUDENT$StudentId	Grade
MIS380	11111	A
MIS400	11111	B
MIS380	22222	B+

Relationships in the Relational Model



RELATIONSHIPS ARE IMPLIED BY THE OVERLAPPING KEYS BETWEEN TABLES.

ONLY KEYS THAT ARE IN THE SAME DOMAIN CAN BE MATCHED IN THIS WAY.

Data Integrity

SINCE THE RELATIONAL MODEL
RELIES HEAVILY ON DUPLICATE KEYS,
THE INTEGRITY (I.E. VALIDITY)
OF THE KEYS IN THE TABLES IS CRITICAL

ENTITY INTEGRITY - EVERY TABLE SHOULD HAVE A PRIMARY KEY THAT CONTAINS UNIQUE, NON-NULL VALUES THAT ARE MINIMAL AND NONUPDATEABLE.

REFERENTIAL INTEGRITY- FOREIGN KEYS MUST CONTAIN VALUES THAT MATCH (IN THE SAME DOMAIN) VALUES IN THE PRIMARY KEY THEY REFER TO (DUPLICATE).

NO UNMATCHED FOREIGN KEY VALUES

DATA INTEGRITY

THERE ARE TWO ENTITY INTEGRITY AND THREE REFERENTIAL INTEGRITY VIOLATIONS IN THE DATA BELOW - CAN YOU FIND THEM?

INSTRUCTOR

InstructorId	FirstName	LastName	Phone
11	JOE	SMITH	593-2736
22	SAM	SLICK	594-4657
22	SALLY	SLY	593-9875
44	SUE	SLIM	593-4756

GRADES

COURSE$CourseCode	**STUDENT$StudentId**	Grade
MIS380	1111	A-
MIS225	1111	B+
MIS225	1111	B
MIS333	2222	A
MIS225	2222	C
MIS380	3444	A

COURSE

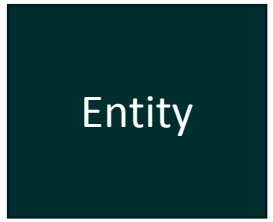
CourseCode	Title	Hours	**INSTRUCTOR$InstructorId**
MIS380	DATABASE I	4	45
MIS225	VISUAL BASIC	4	11
MIS420	SYSTEMS II	4	11

STUDENT

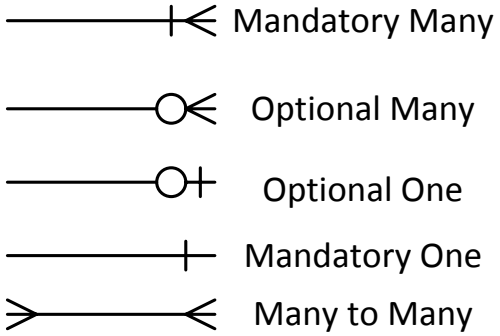
StudentId	FirstName	LastName
1111	JIM	GREEN
2222	STEVE	BLACK
3333	LINDA	BROWN
4444	EMMA	WHITE

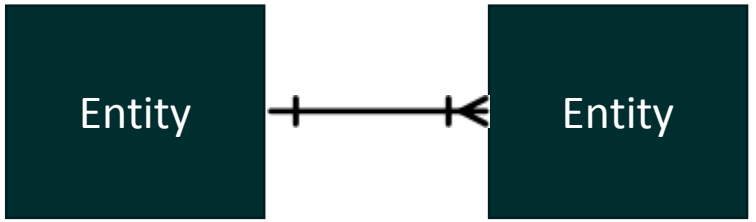
Entity-Relationship Diagram (Review)

- **Entity:** A noun, represents people, places or things that is diagramed as a box that represents a database table



- **Cardinality:** defines the relationship between the entities





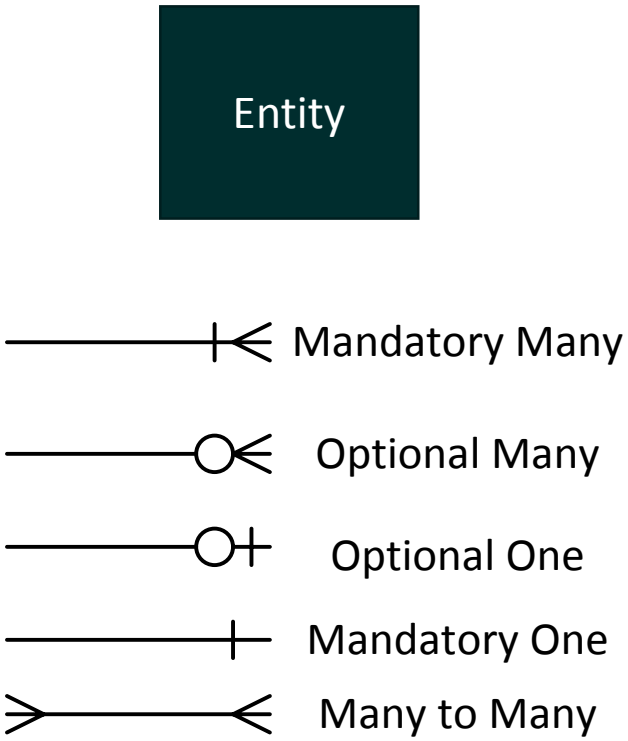
ERD Lab 1

(Small Teams of 3 – 4)

Team 1

Draw the ERD that corresponds with the statement

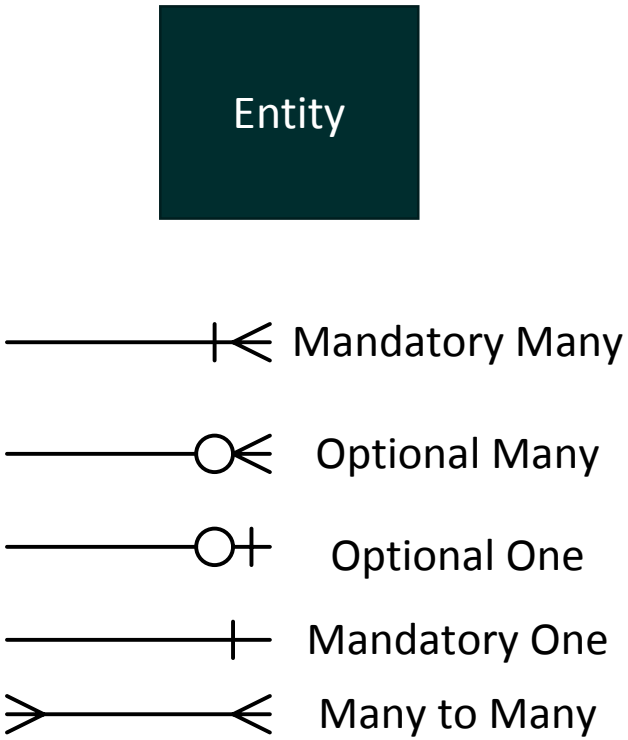
For each product in the warehouse, there are multiple suppliers that can supply the product and we get multiple products from each supplier. For the products, we need to keep track of the product code, description and price. For each supplier, we need to know the name address and phone number.



Team 2

Draw the ERD that corresponds with the statement

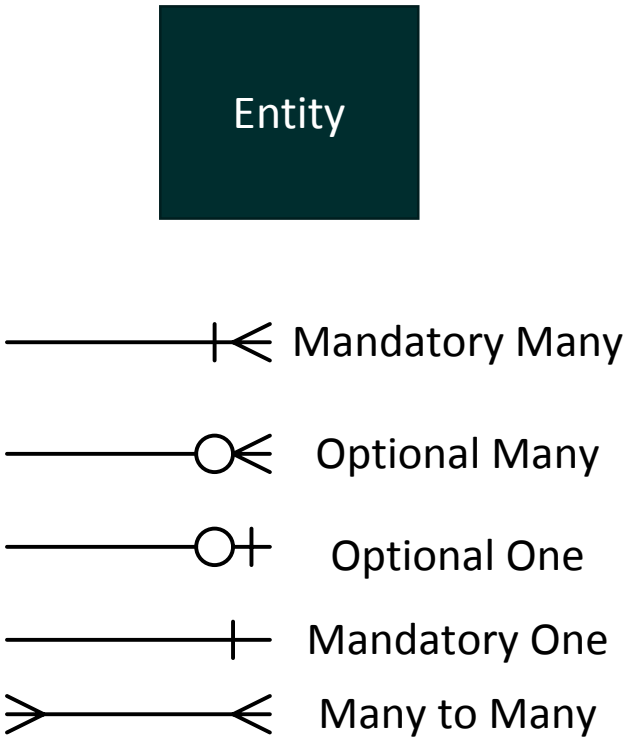
A student rental web site has a database of rental properties and the owners of those properties. Each property has one owner but a particular owner will have multiple properties listed. For each owner, they need to track the name and phone number and for each property, they need the address and the monthly rent.



Team 3

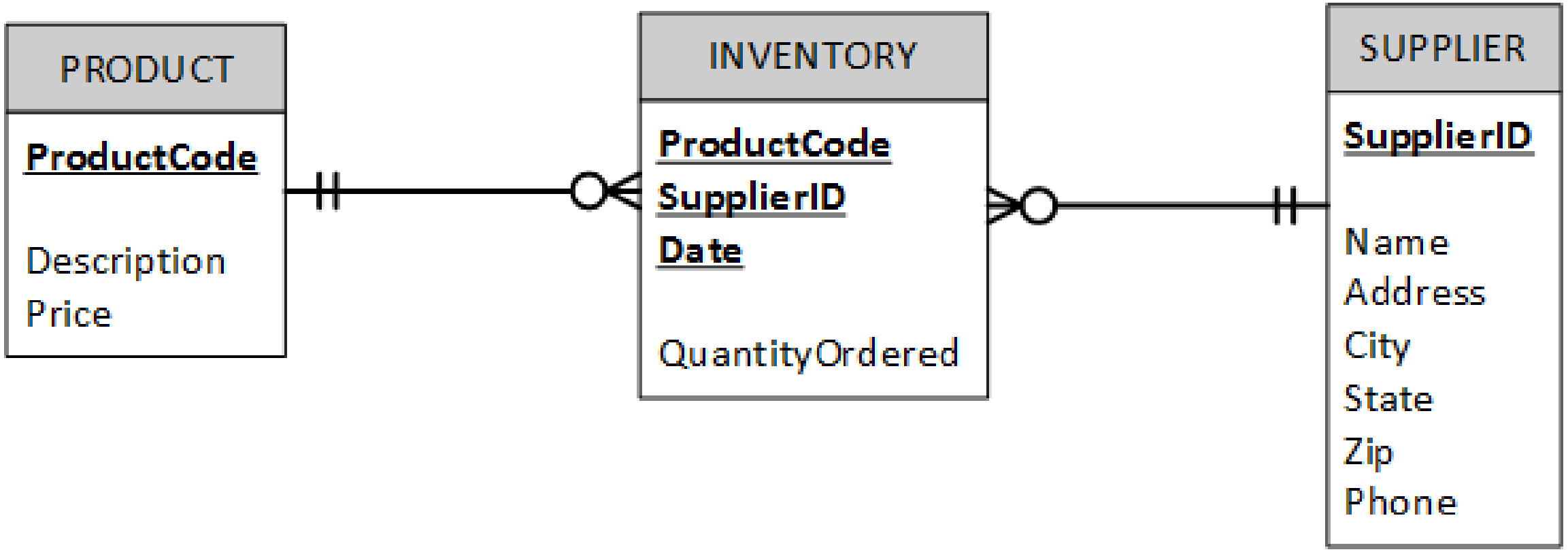
Draw the ERD that corresponds with the statement

An auto repair shop needs to track cars that it services and the various service procedures that have been performed. A particular car will have multiple procedures done on it over time and a particular procedure like an oil change will be performed on many cars. For each car, the shop tracks the VIN number, make, model and year and for each service procedure they have a code, description and price.



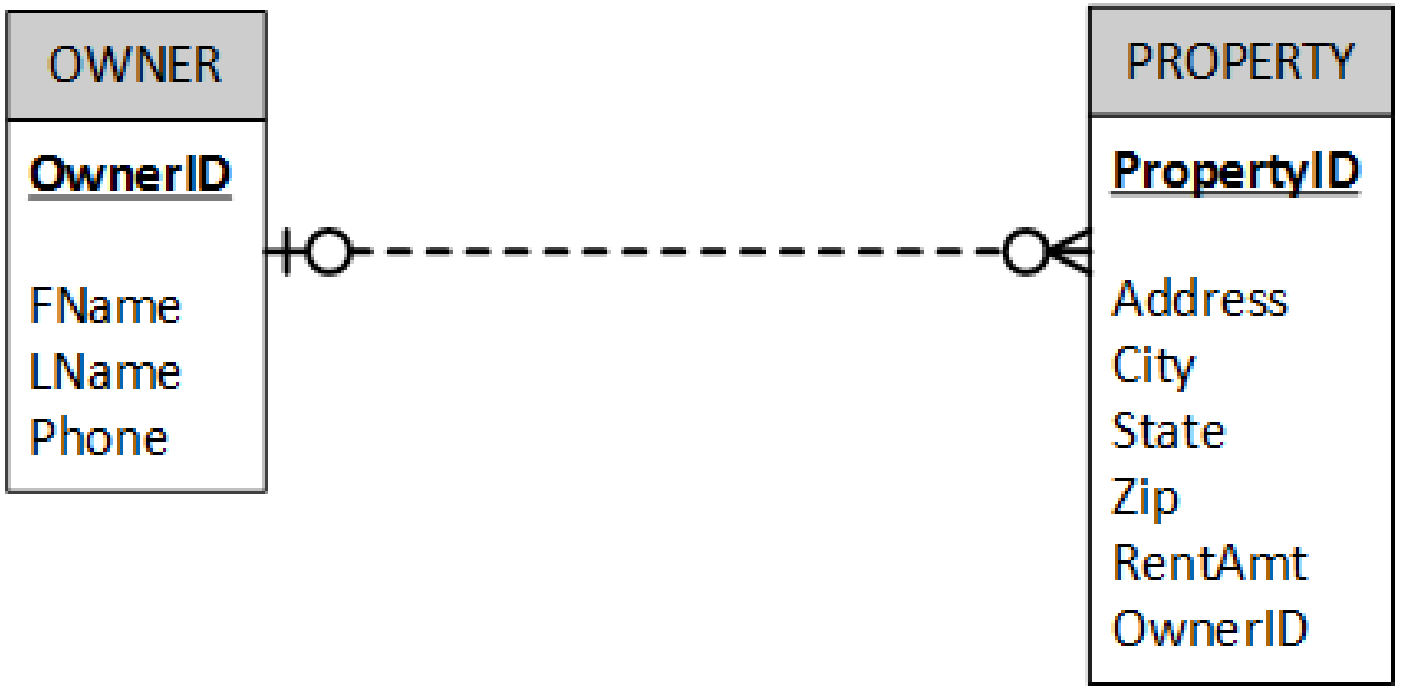
Answers

Team 1



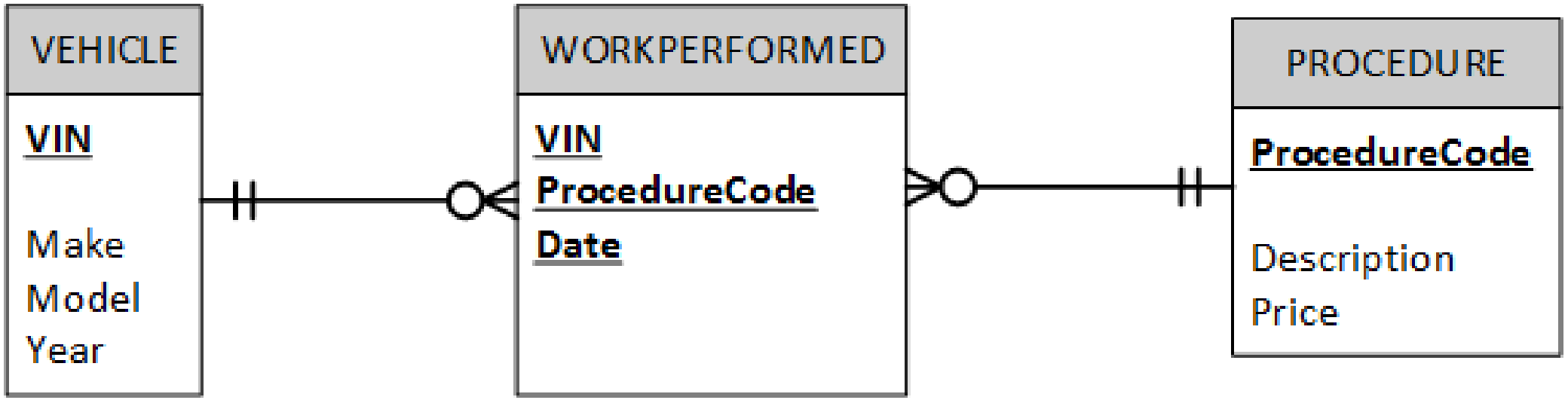
Answers

Team 2



Answers

Team 3



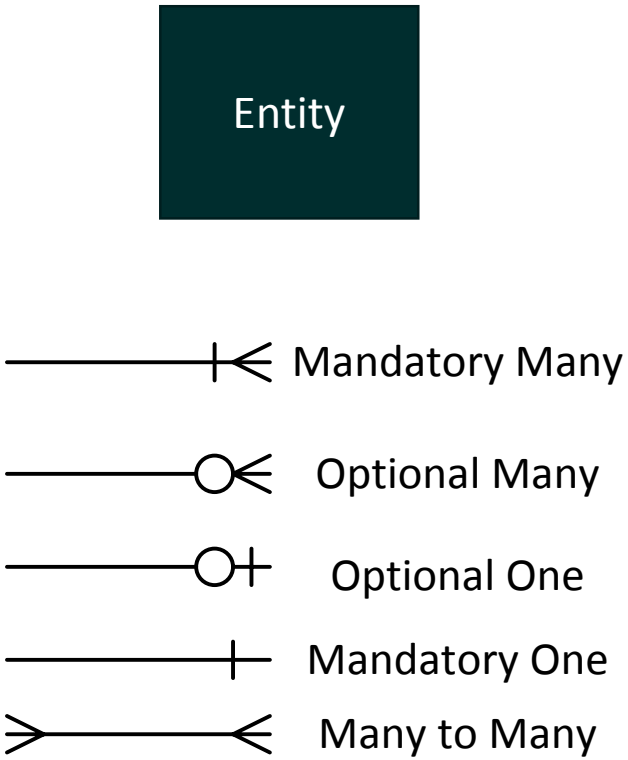
ERD Lab 2

(Individually)

ERD Lab 2 (Project Management)

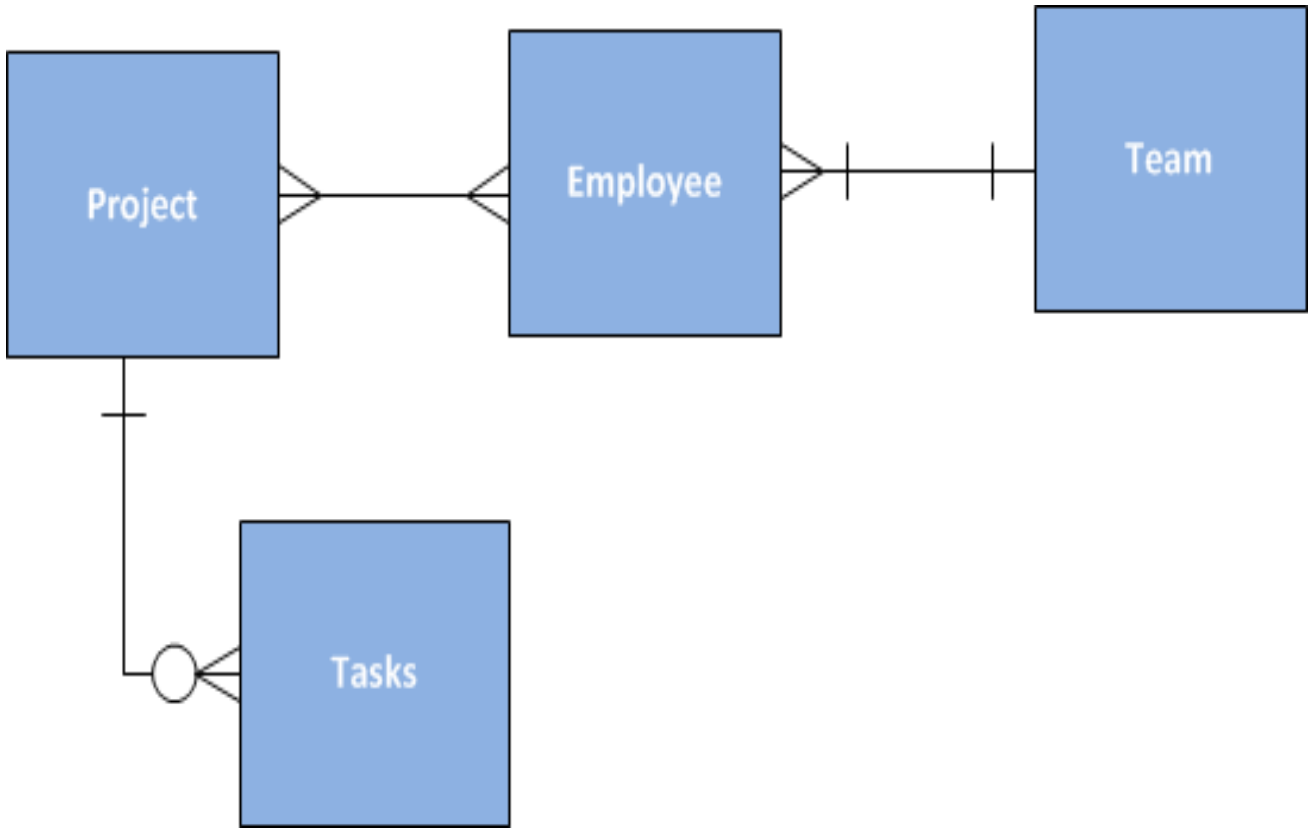
Create an ERD from the list of instructions below

I need a way to keep track of all projects that are assigned to my employees. Sometimes multiple team members can be assigned to one or many projects. Each employee I manage belongs to a team. It would also be nice if my employees could create tasks on the project so they know what work needs to be done. I'll also need a report of all the employees on a team, and what projects they are assigned to or working on.



Exercise 2 solution

Project Management ERD



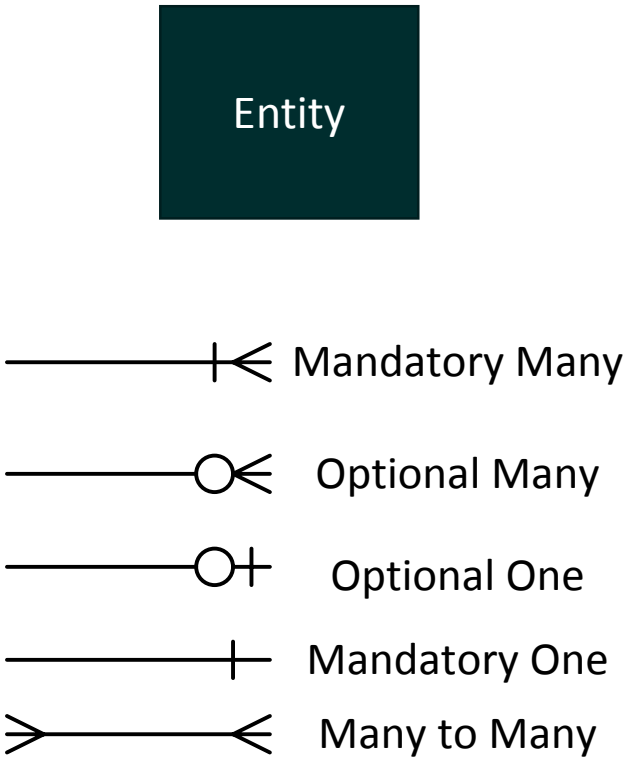
ERD Lab 3

(Individually)

ERD Lab 3 (Contracts Lifecycle Management)

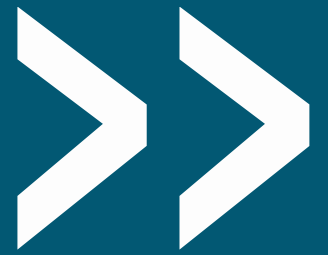
Create an ERD from the list of instructions below

Create a complete Contracts Management Lifecycle ERD to include Contract Requests, Contract Drafting, and Contract Approval & Execution for your client. Must support three types of contracts: Buy Side (Vendor), Sell Side (Customer), Employee Contracts/NDAs. Contracts can be on Client's Paper or on 3rd Party paper.



WorkView Class Design

Relevance to ERDs and Proper Database Design



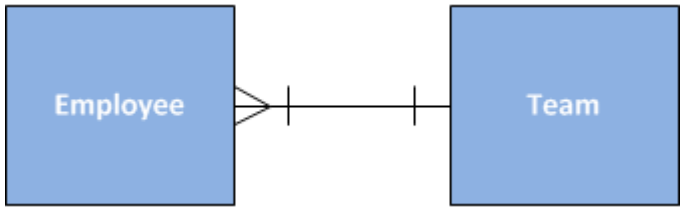
ERD to WorkView

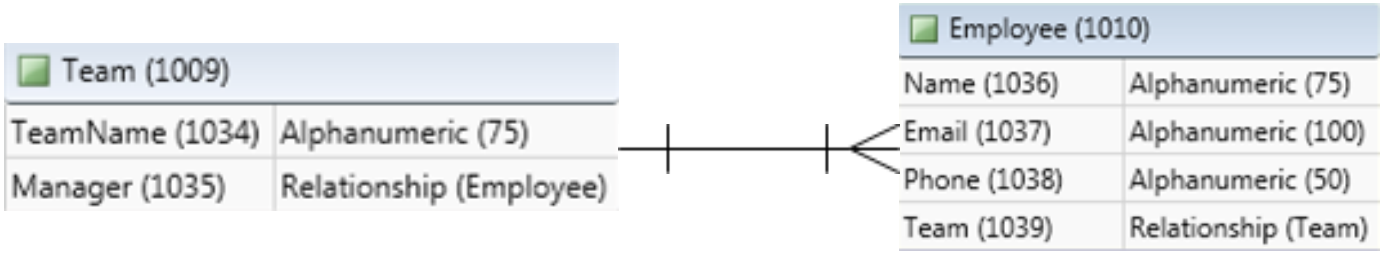
ERD

- Entity
- Attribute
- Cardinality

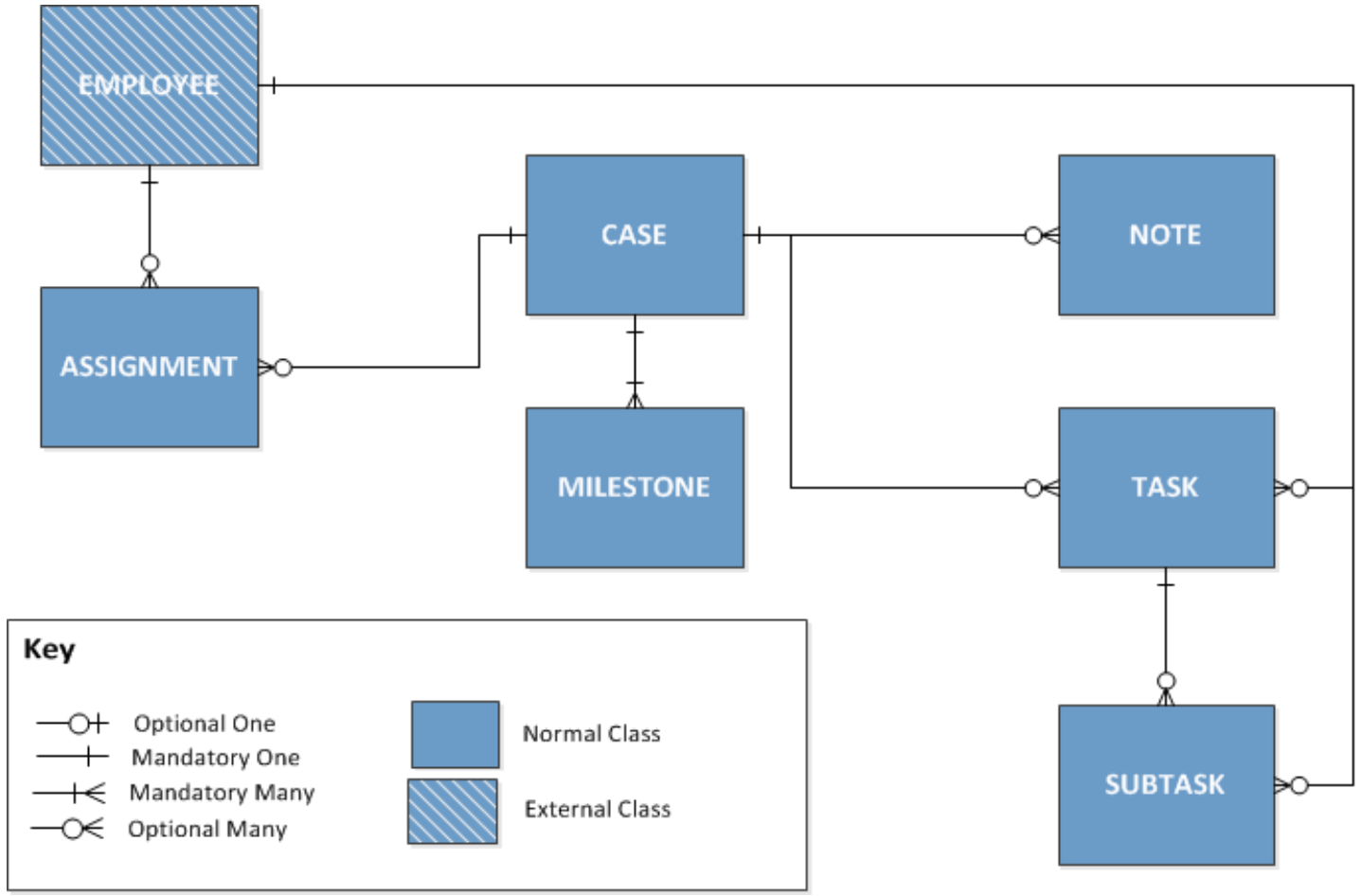
WorkView

- Class
- Attribute
- Relationship attribute





Sample high-level ERD



Class Types

- Standard (aka Normal)
  - Admin defines table and attributes
- Mandatory One
- Optional Many
- External
- Mandatory Many
  - Reference data in another application
    - ODBC, linked server or web service
- Association
  - Facilitates many to many relationships

WorkView Class Types

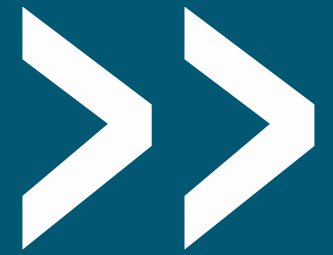
- **External Classes**
  - When to Use
  - Best Practices/Examples
- **Associative Classes**
  - When to Use
  - Best Practices/Examples
- **Extended Classes**
  - When to Use
  - Best Practices/Examples

WorkView Filter Design

Relevance to ERDs and Proper Database Design

Database Administration

Relevance to ERDs and Proper Database Design



Summary

- Solution Design vs. Database Design
- Data Modeling/Design Concepts
- ERD Diagramming Labs
- WorkView Class Design
- WorkView Filter Design
- Database Administration

THANK YOU

- **Thank you for attending my course!**
- **Thank you for attending TechQuest – July 2015!**
- **I Value Your Feedback!**
  - *I welcome your feedback/comments! Please complete the course evaluation and any TechQuest evaluations.*
  - *Enjoy the remainder of TechQuest and your courses!!*