# 1.264 Lecture 9

Data modeling II

This class: Hand in exercises electronically after class
Please start Visual Paradigm
Next class: No reading. Exercises due <u>after</u> class

Keys

- **Primary key: one or more attributes that uniquely identify a record.**
  - **What would you use in a customer database of 100,000 people and no unique customer id?**
    - **Name not unique**
    - **Add birthdate, but not guaranteed to be unique**
    - **Address can change**
    - **Can use social security number, but not everyone has one**
    - **Privacy is an issue**
  - **Issues in choosing a primary key**
    - **Stability**
    - **Control**
    - **Use a system generated key if possible in many cases**
  - **Which is better: email address or customer ID?**

Foreign keys

- **Primary key of the <u>independent</u> or parent entity type is maintained as a non-key attribute in the <u>dependent</u> or child entity type**
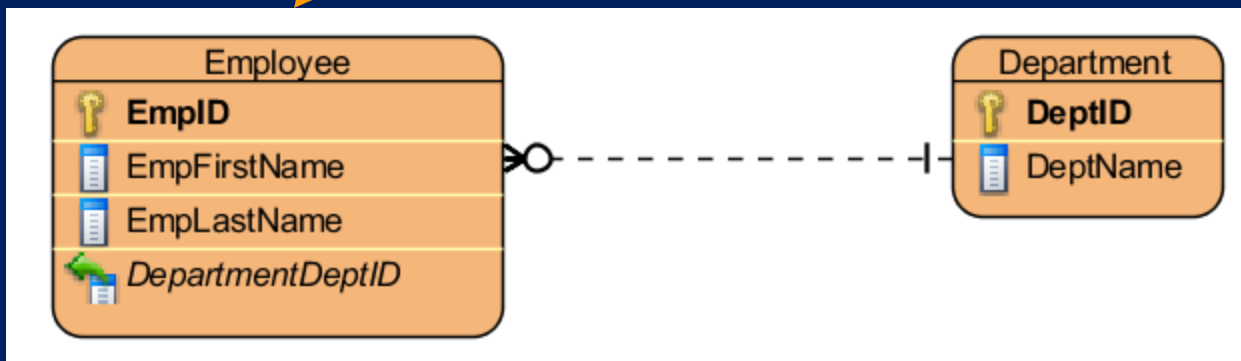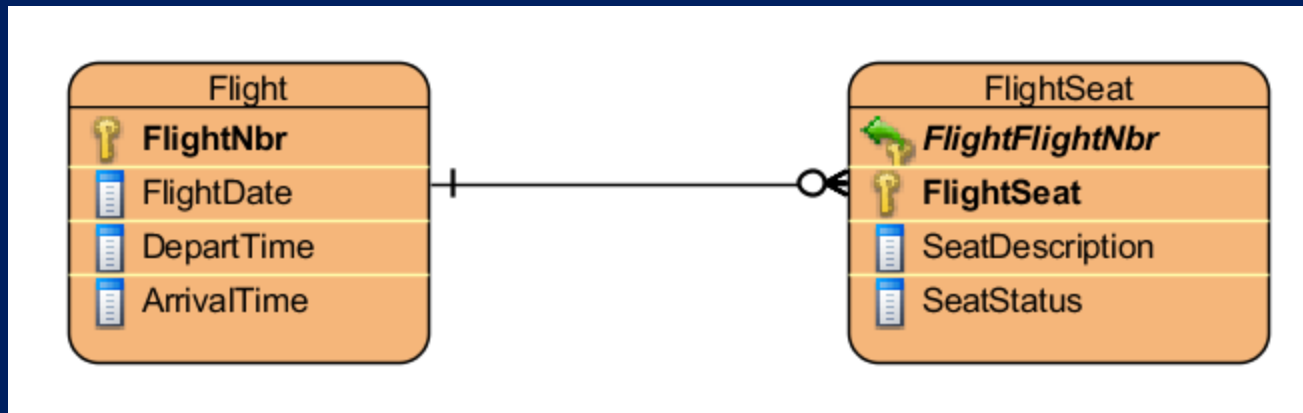
Foreign keys
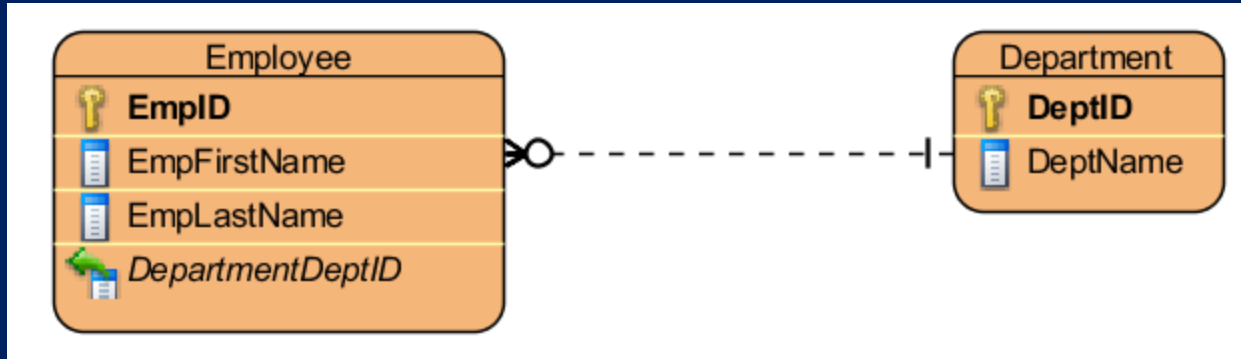
EmpID	DeptID	EmpLastName	EmpFirstName
4436	483	Brown	John
4574	483	Jones	Helen
5678	372	Smith	Jane
5674	372	Crane	Sally
9987	923	Black	Joe
5123	923	Green	Bill
5325	483	Clinton	Bob

DeptID	DeptName
930	Receiving
378	Assembly
372	Finance
923	Planning
483	Construction

**Database requires a valid department number when employee is added
Employee ID is the unique identifier of employees; department number
is not needed as part of the employee primary key**
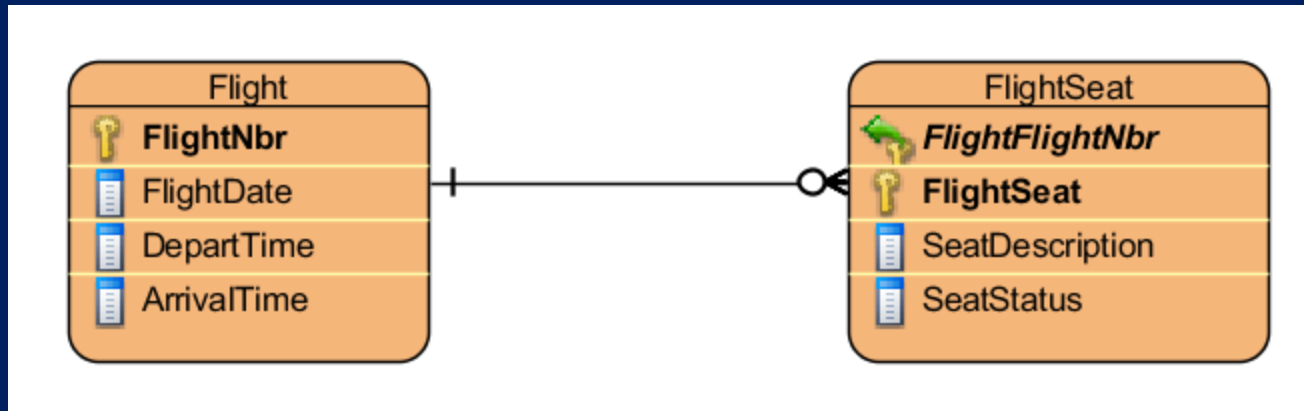
Composite foreign keys

Independent/parent

Dependent/child (must contain, as a foreign key, the primary key of the independent entity)

Assume a charter airline: every flight has a different number
**What has to change if this is a scheduled carrier?**

Composite foreign keys

Flight			
FlightNbr	**FlightDate**	**DepartTime**	**ArrivalTime**
243	9/24/00	9:00am	11:00am
253	9/24/00	10:00am	12:30pm
52	9/24/00	11:00am	2:00pm

FlightSeat			
FlightNbr	**FlightSeat**	**SeatStatus**	**SeatDescription**
243	8A	Confirmed	Window
243	7D	Reserved	Aisle
243	14E	Open	Center
253	1F	Open	Window
253	43A	Confirmed	Window

Flight number must be part of the flight seat primary key; this is different than employee and department, where department is not required.
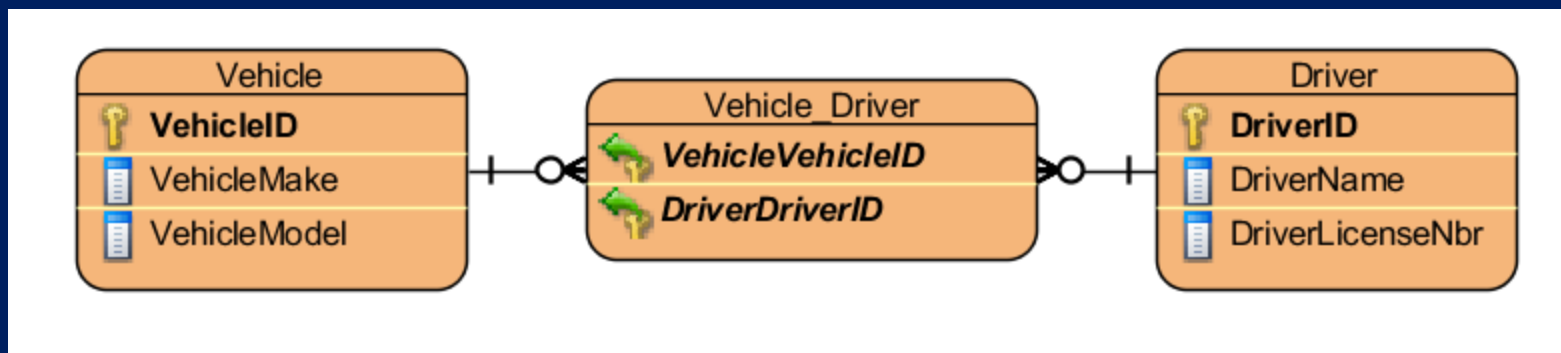
Foreign keys (many-many relationships)

- **Primary key of parent is used in primary key of child**

Independent   **Dependent**   **Independent**
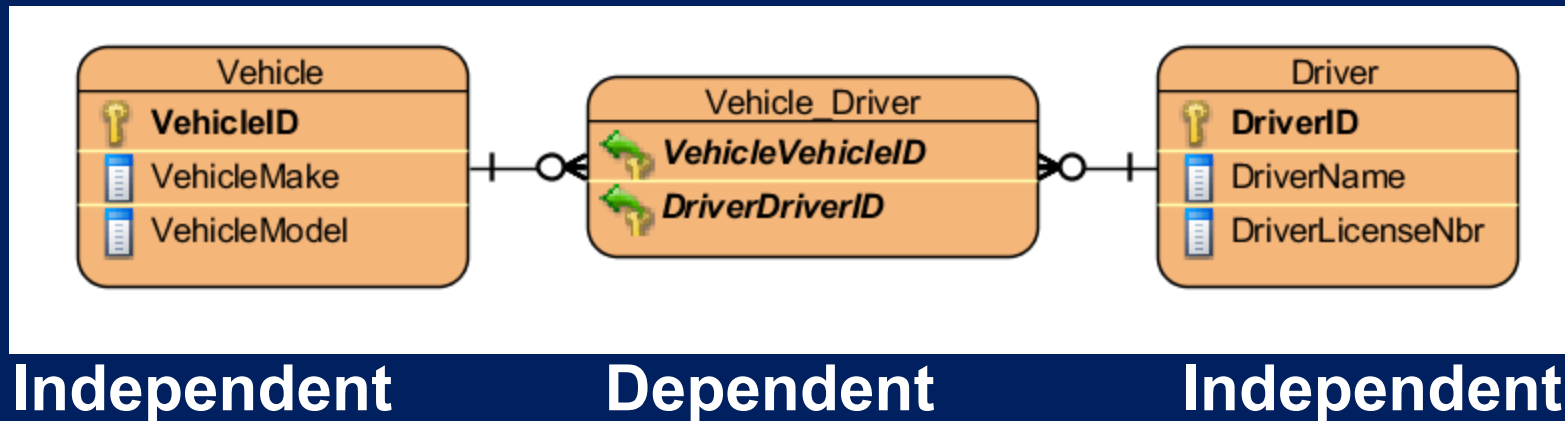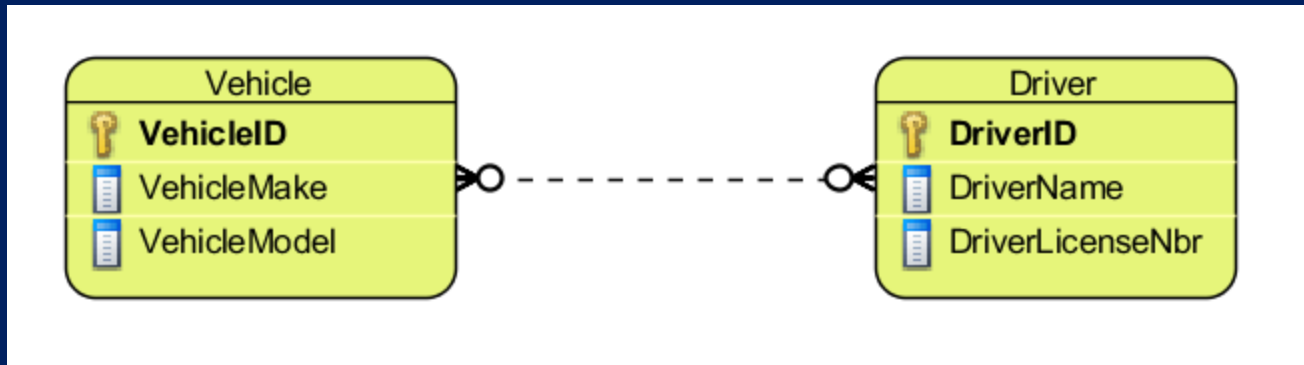
Vehicle can be driven by many drivers; driver can drive many vehicles

Show in Visual Paradigm. Set presentation options.

Many-to-many relationships with foreign keys

Vehicle		
VehicleID	VehicleMake	VehicleModel
35	Volvo	Wagon
33	Ford	Sedan
89	GMC	Truck

Vehicle Driver	
VehicleID	DriverID
35	900
35	253
89	900

Driver		
DriverID	DriverName	DriverLicenseNbr
253	Ken	A23423
900	Jen	B89987

Never create an entity with vehicle1, vehicle2,… !

Referential integrity

- **Referential integrity maintains the validity of foreign keys when the primary key in the parent table changes. (The database software does this.)**
  - **Every foreign key either matches a primary key (or is null)**
  - **E.g., you cannot add an employee to an invalid department (or, usually, a null department)**
- **Cascade rules. Choose among two delete options:**
  - **Cascade restrict: Rows in the primary key table can't be deleted unless all corresponding rows in the foreign key tables have been deleted.**
    - **E.g., when deleting a department, don't delete all the employees**
  - **Cascade delete: When rows in the primary key table are deleted, associated rows in foreign key tables are also deleted**
    - **E.g., when deleting an order, delete all items in the order**
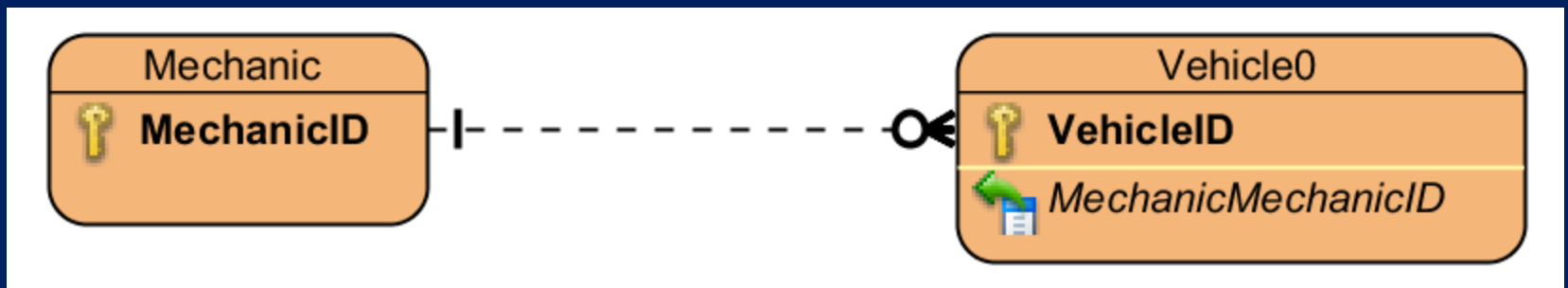
MIT Trucking Company

- **Exercise**
  - **Determine and name entity types**
  - **Determine cardinality (1:N, N:N) and type (mandatory/optional) of relationships**
  - **Add identifiers and primary and foreign keys**
  - **Label relationship phrases (if you have time)**
- **Use Visual Paradigm:**
  - **Create new entities (toolbar)**
  - **Set attributes and primary keys (properties)**
  - **Let the relationships fill in the foreign keys**
  - **Ignore data type (accept integer default)**
  - **Edit the relationship to be 1-many, 0/1-many if you have time**
  - **Four exercises follow: Use separate diagrams for each**

MIT Trucking Company exercise 1

- **Trucking company employs mechanics to maintain vehicles. Each mechanic usually assigned to many vehicles. Vehicle always assigned to 1 mechanic.**
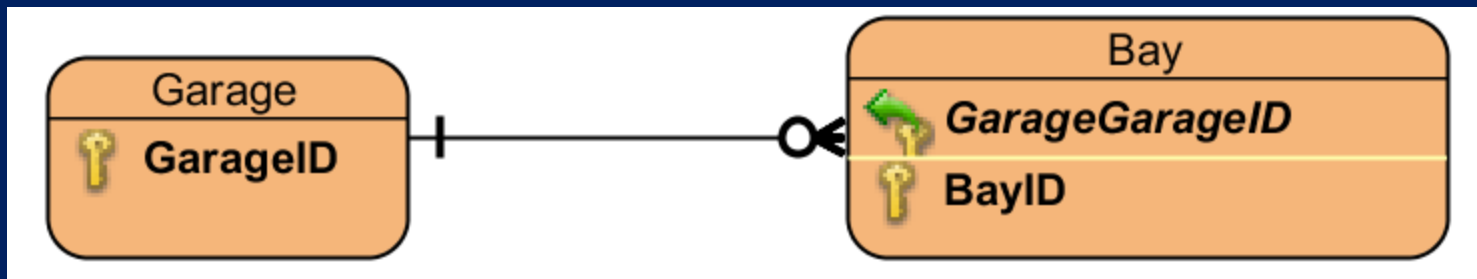
MIT Trucking Company solution 1

- **Trucking company employs mechanics to maintain vehicles. Each mechanic usually assigned to many vehicles. Vehicle always assigned to 1 mechanic.**

MIT Trucking Company exercise 2

- **Trucking company has several garages. A garage may contain many bays. A bay must be in a garage.**

MIT Trucking Company solution 2

- **Trucking company has several garages. A garage may contain many bays. A bay must be in a garage.**

MIT Trucking Company exercise 3

- **Garage employs managers to monitor vehicle availability and repair costs. Each manager is assigned at least 1 and usually many vehicles. A vehicle may or may not have a manager responsible.**

- **Use Vehicle2 as entity**

MIT Trucking Company solution 3

- **Garage employs managers to monitor vehicle availability and repair costs. Each manager is assigned at least 1 and usually many vehicles. A vehicle may or may not have a manager responsible.**
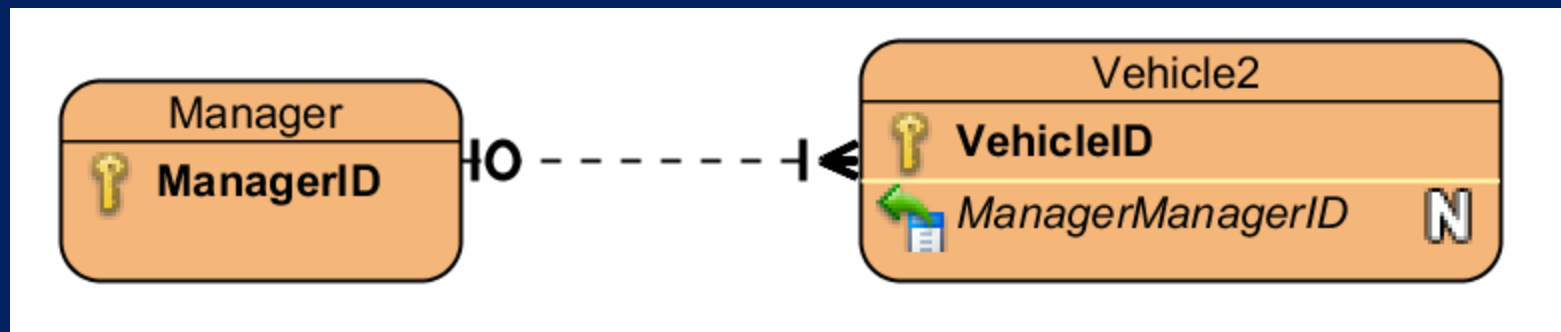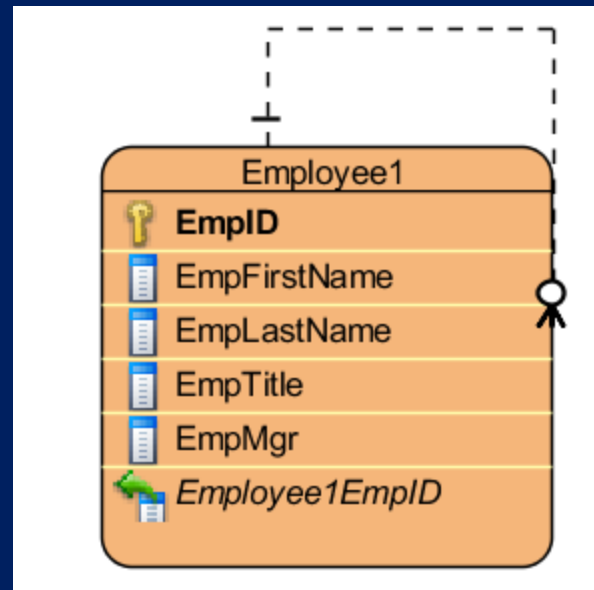
MIT Trucking Company exercise 4

- **Garage maintains a list of specialized repair personnel who are used as necessary. A specialist may work on many vehicles. A vehicle may or may not be repaired by a specialist.**

- **Use Vehicle3 as entity**

MIT Trucking Company solution 4

- **Garage maintains a list of specialized repair personnel who are used as necessary. A specialist may work on many vehicles. A vehicle may or may not be repaired by a specialist.**
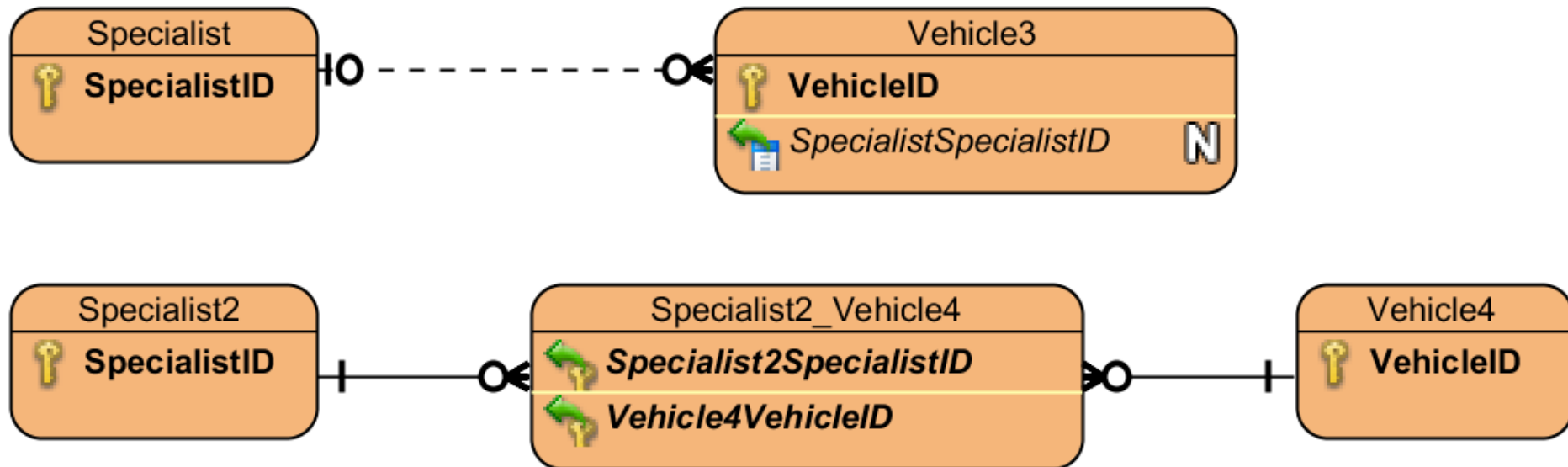
Recursive relation

- **We'll cover this in more detail under SQL. Pretend the recursive relation is between two tables, the real one and a virtual copy. In this case, a manager table and an employee table. Proceed as usual, with a small syntax change.**

1.264J / ESD.264J Electric Machines
Fall 2013