Lecture 7

Data Modeling

1

Data Modeling

- Data modeling is the act of exploring data-oriented structures.
- During the process, business rules and requirements are revealed.
- From the point of view of an object-oriented developer, data modeling is conceptually similar to class modeling.
  - With data modeling you identify entity types whereas with class modeling you identify classes.
  - Data attributes are assigned to entity types just as you would assign attributes and operations to classes.
  - There are associations between entities, similar to the associations between classes – relationships, inheritance, composition, and aggregation are all applicable concepts in data modeling.

2

Data Models

- **Conceptual/Logical data models (LDMs)**.
  - Explore the domain concepts and their relationships.
  - LDMs depict the logical entity types (typically referred to as entity types or entities), the data attributes describing those entities, and the relationships between the entities.
- **Physical data models (PDMs)**.
  - PDMs are used to design the internal schema of a database, depicting the data tables, the data columns of those tables, and the relationships between the tables.

3

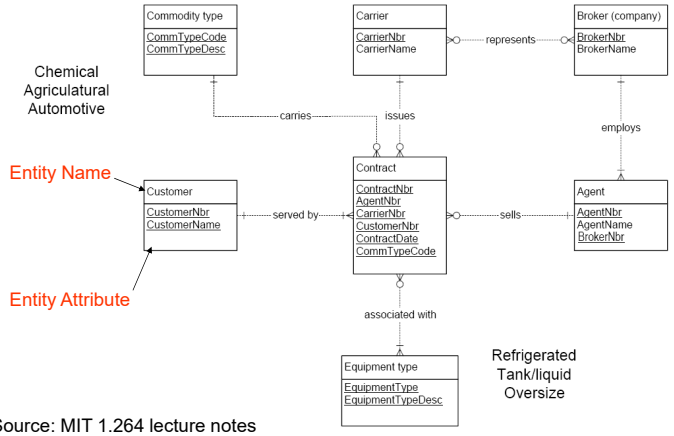
Logical Data Modeling

- Method to discover the data, relationships and rules of a business, collectively called the business rules
- Logical data models are the basis of:
  - Physical data models, or actual databases
  - Applications, parts of which can be automatically generated from the data model
- Small model for EZ Broker of transportation services
  - Small, but says a lot about EZ Broker
  - Gives good picture of what database should look like
  - Also gives good picture of underlying business rules of broker
    - Useful in requirements analysis and scrubbing!

4

Source: MIT 1.264 lecture notes

EZ Broker Data Model



Source: MIT 1.264 lecture notes

EZ Broker Business Rules

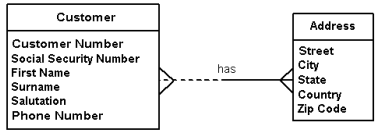
- A carrier can be associated with many brokers
- A broker can be associated with many carriers
- A carrier can issue many contracts
- A contract is issued by one carrier
- A broker can employ many agents
- An agent is employed by one broker
- An agent can sell many contracts
- A contract is serviced by only one agent
- A contract can serve to carry only one commodity type
- A commodity type can be carried under many contracts
- A contract can be associated with many equipment types
- An equipment type can be associated with many contracts
- A customer can be served by many contracts
- A contract covers one customer

Source: MIT 1.264 lecture notes

Data Model Purpose

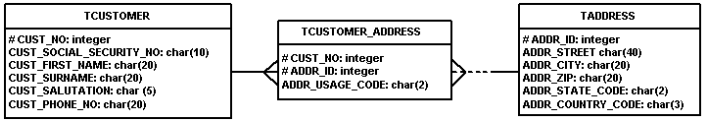
- Business needs to build logical data model so users and developers both understand business rules of the company
  - Models enable users and developers to have a single view of the system
  - Sometimes users note this is first time they understood business rules!
  - Class modeling is small extension
    - Not only model data, but also the methods (procedures) that operate on each
- Converting logical to physical data model (database) is very straightforward.
  - e.g. converting many-to-many relations …

Logical Data Model



These diagrams are called ER diagrams

Physical Data Model



Source: http://www.agiledata.org/essays/dataModeling101.html

Steps in Data Modeling

- Identify entities
- Identify attributes
- Apply naming conventions
- Identify relationships
- Apply data model patterns
- Assign keys
- Normalize to reduce data redundancy
- Denormalize to improve performance

This steps are performed iteratively until satisfactory.

9

Identify Entities (1/2)

- An entity type, also simply called "entity", is similar conceptually to OOP's concept of a class
  - e.g. *people, places, things, events, cars*
  - In an order entry system
    - *Customer*, *Address*, *Order*, *Item*, *Tax, ….*
  - Entities are things, often physical, that have facts associated with them.
  - Processes are almost never entities
    - e.g., order entry is not an entity
    - Orders and customers are entities
    - Reports are not entities
- Entity type descriptions should be as extensive as possible in developing a model.

10

Identify Entities (2/2)

- Poor description
  - Vendor: Someone we buy products from.
- Good description
  - Vendor: A US corporation we have reviewed with respect to their qualifications for providing products to our company. Vendors are rated based on price, quality, delivery performance and financial stability. Each vendor is classified by one vendor status: approval pending, approved, rejected or inactive. This approval decision is made in a weekly meeting among purchasing, manufacturing and finance. Purchasing requests that rejected vendors be kept in the database for future reference. Purchasing expects 500 vendors will be maintained at any one time. Of this, 200 will be active, 25 pending, 75 inactive and 100 rejected. Contact Joan Smith in Purchasing for more information.

11

Identify Attributes & Use Naming Convention

- Each entity type will have one or more data attributes,
  - People ← firstName, lastName, bloodType, weight, height, …
  - Attributes are mostly nouns
- Attributes should also be cohesive from the point of view of your domain, something that is often a judgment call.
  - e.g. 1: People ← name
  - e.g. 2: People ← firstName, lastName
  - Bad example: People ← orderNumber
- Naming convention provides guide lines for naming entities and attributes.
  - the logical naming conventions should be focused on human readability: people ← firstName, lastName
  - the physical naming conventions will reflect technical considerations: people ← sFirstName, sLastName

12

Entity Type/Attribute Exercise

- Instructor
- Student
- Course section number
- Building name
- Course number
- Textbook price
- Student name
- Instructor ID
- Textbook author
- Course title
- Textbook
- Classroom
- Textbook ISBN
- Section days
- Office hours
- Textbook title
- Classroom number
- Student ID
- Instructor name
- Textbook publisher
- Section capacity
- Course objective
- Copyright date
- Building number
- Course section
- Course
- Building
- Section time
- Classroom capacity

Section no.: 第 no 次上課
ISBN: 圖書編號
Office hour: 課外與老師諮詢時間
objective: 目的

13

Source: MIT 1.264 Lecture Notes

Exercise

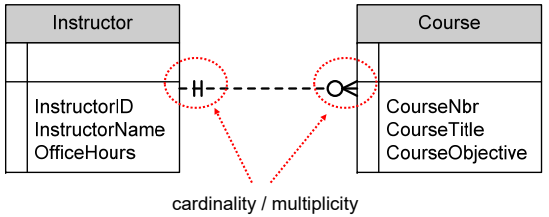
Please find out entities and their attributes shown in the previous slide and draw entities representations on a worksheet (10 minutes)

14

Source: MIT 1.264 Lecture Notes

Identify Relationships (1/3)

- Relationships are lines between boxes (entities)
- Cardinality is the expected number of related occurrences between the two entities in the relationship
- Relationships + cardinality = business rules

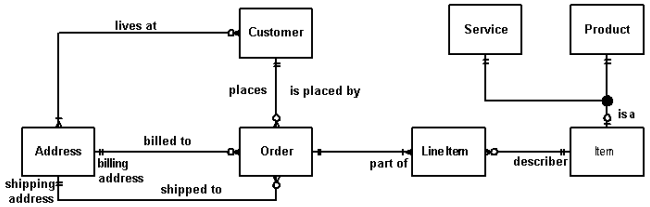


cardinality / multiplicity

15

Identify Relationships (2/3)

- Entities have relationships with other entities.
  - Customers PLACE orders
  - Customers LIVE AT addresses



16

Source: http://www.agiledata.org/essays/dataModeling101.html

Identify Relationships (3/3)
Attributes of Relationships

- Cardinality
  - One to one (1:1); E.g. one person has exactly one name.
  - One to many (1:n); E.g. one person may have one or many cars.
  - Many to many (m:n); E.g. One student can take courses from many teachers, and one teacher may teach many students.

- Identifying vs. Non-identifying
  - Identifying
    - E.g. One building has many rooms, and these rooms cannot exist without the building. (parent – children, you can identify an individual parent from children)
  - Non-identifying
    - E.g. One DVD may be rented by many customers, you cannot identify a DVD from customers because customers may have rented lots of CDs

17

Assign Keys (1/3)

- A key is data attribute(s) that uniquely identify an entity.
  - A key with two or more attributes is called a *composite key*.
  - E.g. ID numbers (student ID, SSN, …)
  - A key that is formed by existing attributes in the real world is called a *natural key*.

- Exercise: What is/are good data attribute/attributes for keys of the following two entities?
  - Assume a car entity with the following attributes
    - Plate number, manufacturer, type, color, owner, owner phone number, owner address
  - Assume a employee entity has the following attributes
    - First name, last name, phone number, address, blood type

18

Assign Keys (2/3)

- A key (attribute) must be able to uniquely identify an physical entity (a row in the physical database table)
  - We need to be able to find a specific car from many cars. This cannot be achieved by specifying color, thus color cannot be used as a key
  - By specifying plate number, we should be able to select a single car out of many cars. Thus, plate number can be used as a key.
- The selected key for an entity is called "primary key".
- Assuming related entities A and B share an attribute K, which is the primary key for entity B. Then attribute K becomes a "foreign key" of entity A.

19

Assign Keys (3/3)
Selection Strategy

1. Use natural key
   - Natural keys are existing attributes, thus no additional data needs to be introduced in the data schema.
   - They have business meaning, and it is possible that they may need to change if your business requirement change. → keys are like "glue" between tables, and changing keys changes the coupling between different tables.
   - E.g. Name for people
2. Use surrogate key
   - A new artificial data attribute that has no business meaning
   - Incremental Keys: A database assigned numerical ID (autoNumber in Access) that increments as new rows are created
   - UUID: a 128-bit hash value from Ethernet card MAC address & the current date and date of the computer
   - GUID: A Microsoft standard that extends UUID
   - E.g. ID Number for people

20

Steps in Data Modeling

- Identify entities
- Identify attributes
- Apply naming conventions
- Identify relationships
- *Apply data model patterns*
- Assign keys
- Normalize to reduce data redundancy
- Denormalize to improve performance

This steps are performed iteratively until satisfactory.

Identify Attributes & Use Naming Convention

- Each entity type will have one or more data attributes,
  - People ← firstName, lastName, bloodType, weight, height, …
  - Attributes are mostly nouns
- Attributes should also be cohesive from the point of view of your domain, something that is often a judgment call.
  - E.g. 1: People ← name
  - E.g. 2: People ← firstName, lastName
  - Bad example: People ← orderNumber
- Naming convention provides guide lines for naming entities and attributes.
  - the logical naming conventions should be focused on human readability: people ← firstName, lastName
  - the physical naming conventions will reflect technical considerations: people ← sFirstName, sLastName

Entity Type/Attribute Exercise

- **Instructor**
- Student
- Course section number
- Building name
- Course number
- Textbook price
- Student name
- Instructor ID
- Textbook author
- Course title
- Textbook
- Classroom
- Textbook ISBN
- Section days
- Office hours
- Textbook title
- Classroom number
- Student ID
- Instructor name
- Textbook publisher
- Section capacity
- Course objective
- Copyright date
- Building number
- Course section
- Course
- Building
- Section time
- Classroom capacity

Modeling school life…

Section no.: 第 no 次上課
ISBN: 圖書編號
Office hour: 課外與老師諮詢時間
objective: 目的

Source: MIT 1.264 Lecture Notes

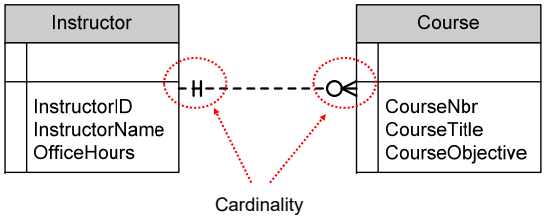
Exercise

Please find out entities and their attributes shown in the previous slide and draw entities representations on a worksheet (10 minutes)

Source: MIT 1.264 Lecture Notes

Identify Relationships (1/3)

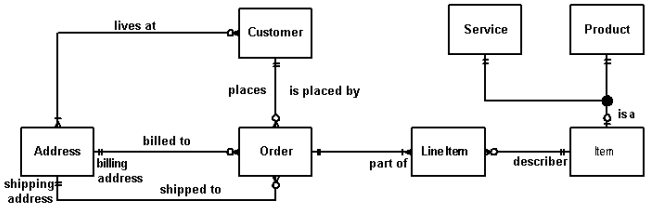
- Relationships are lines between boxes (entities)
- Cardinality is the expected number of related occurrences between the two entities in the relationship
- Relationships + cardinality = business rules



25

Identify Relationships (2/3)

- Entities have relationships with other entities.
  - Customers PLACE orders
  - Customers LIVE AT addresses



26

Source: http://www.agiledata.org/essays/dataModeling101.html

Identify Relationships (3/3)
Attributes of Relationships

- Cardinality
  - One to one (1:1); E.g. one person has exactly one name.
  - One to many (1:n); E.g. one person may have one or many cars.
  - Many to many (m:n); E.g. One student can take courses from many teachers, and one teacher may teach many students.
- Identifying vs. Non-identifying
  - Identifying
    - E.g. One building has many rooms, and these rooms cannot exist without the building. (parent – children, you can identify an individual parent from children)
  - Non-identifying
    - E.g. One DVD may be rented by many customers, you cannot identify a DVD from customers because customers may have rented lots of CDs

27

Assign Keys (1/3)

- A key is data attribute(s) that uniquely identify an entity.
  - A key with two or more attributes is called a *composite key*.
  - E.g. ID numbers (student ID, SSN, …)
  - A key that is formed of attributes that already exist in the real world is called a *natural key*.
- Exercise: What is/are good data attribute/attributes for keys of the following two entities?
  - Assume a car entity with the following attributes
    - Plate number, manufacturer, type, color, owner, owner phone number, owner address
  - Assume a employee entity has the following attributes
    - First name, last name, phone number, address, blood type

28

Assign Keys (2/3)

- A key (attribute) must be able to uniquely identify an physical entity (a row in the physical database table)
  - We need to be able to find a specific car from many cars. This cannot be achieved by specifying color, thus color cannot be used as a key
  - By specifying plate number, we should be able to select a single car out of many cars. Thus, plate number can be used as a key.
- The selected key for an entity is called "primary key".
- Assuming related entities A and B share an attribute K, which is the primary key for entity B. Then attribute K becomes a "foreign key" of entity A.

Assign Keys (3/3)
Selection Strategy

1. Use natural key
   - Natural keys are existing attributes, thus no additional data needs to be introduced in the data schema.
   - They have business meaning, and it is possible that they may need to change if your business requirement change. → keys are like "glue" between tables, and changing keys changes the coupling between different tables.
   - E.g. Name for people
2. Use surrogate key
   - A new artificial data attribute that has no business meaning
   - Incremental Keys: A database assigned numerical ID (autoNumber in Access) that increments as new rows are created
   - UUID: a 128-bit hash value from Ethernet card MAC address & the current date and date of the computer
   - GUID: A Microsoft standard that extends UUID
   - E.g. ID Number for people

Normalization & Denormalization

- Normalization is the process of reducing redundant data
  - Prevents data inconsistency and update anomalies
  - Avoids storing identical data in multiple tables
- Normalization slightly degrades database performance
  - More impacts on reads
  - Little impact on writes, which tend to be the bottleneck anyway
  - Denormalization is common on read-only databases on which high performance is required (e.g. Web read-only databases)
  - Database design and disk configuration (architecture) interact

Some Definitions

- Row or record: a fixed tuple (set) of attributes (fields) that describes an instance of an entity
- Key: a unique identifier for a row in a table, used to select the row in queries. It can be composed of several fields.
- Non-key: all the other fields in the row
- Entity: Object defined in system model about which data is stored in the database. A table in a relational database.

Source: MIT 1.264 lecture notes

Five Normal Forms

1. All occurrences of an entity must contain the same number of attributes. No lists.
2. All non-key fields must be a function of the key.
3. All non-key fields must not be a function of other non-key fields.
4. *A row should not contain two or more independent multi-valued facts about an entity.*
5. *A record cannot be reconstructed from several smaller record types.*

33

Source: MIT 1.264 lecture notes

1st Normal Form

All occurrences of an entity must contain the same number of attributes. No lists.

- no repeating groups
- entries be uniquely identifiable (i.e. must have key!)

Customer	Day	Amount
Pete	Monday	19.00 -28.20
Pete	Wednesday	-84.00
Sarah	Friday	100.00 150.00 -40.00

Customer	Day	Amount
Pete	Monday	19.00
Pete	Monday	-28.20
Pete	Wednesday	-84.00
Sarah	Friday	100.00
Sarah	Friday	150.00
Sarah	Friday	-40.00

34

http://en.wikipedia.org/wiki/First_normal_form

Customer	Day	Amount
Pete	Monday	19.00
Pete	Monday	-28.20
Pete	Wednesday	-84.00
Sarah	Friday	100.00
Sarah	Friday	150.00
Sarah	Friday	-40.00

Person	Foods Not Eaten
Jim	Liver, Goat's cheese
Alice	Broccoli
Norman	Pheasant, Liver, Peas

Customer	Day	Amount
Pete	Monday	19.00
Pete	Monday	19.00

Transaction ID	Customer	Day	Amount
1	Pete	Monday	19.00
2	Pete	Monday	19.00

35

2nd Normal Form (1/2)

All non-key fields must be a function of the full key

- **Example that violates second normal form**
  - Key is Part + Warehouse
  - Someone found it convenient to add Address, making reports easier
  - WarehouseAddress is a fact about Warehouse, not about Part
- **Problems**
  - Warehouse address is repeated in every row that refers to a part stored in a warehouse
  - If warehouse address changes, every row referring to a part stored in that warehouse must be updated
  - Data might become inconsistent, with different records showing different addresses for the same warehouse
  - If at some time there were no parts stored in the warehouse, there may be no record in which to keep the warehouse's address.

Part	Warehouse	Quantity	WarehouseAddress
42	Boston	2000	24 Main St.
333	Boston	1000	24 Main St.
390	New York	3000	99 Broad St.

36

Source: MIT 1.264 lecture notes

2nd Normal Form (2/2)

- **Solution**
  - Two entity types: Inventory, and Warehouse
  - Advantage: solves problems in the last slide
  - Disadvantage: If application needs address of each warehouse stocking a part, it must access two tables instead of one.

Part	Warehouse	Quantity
42	Boston	2000
333	Boston	1000
390	New York	3000

Warehouse	WarehouseAddress
Boston	24 Main St.
New York	99 Broad St.

37

Source: MIT 1.264 lecture notes

3rd Normal Form (1/2)

Non-key fields cannot be a function of other non-key fields

- **Example that violates third normal form**
  - Key is employee
  - Someone found it convenient to add department location for a report
  - Department location is a function of department, which is not a key
- **Problems**
  - Department location is repeated in every employee record
  - If department location changes, every record with it must be changed
  - Data might become inconsistent
  - If a department has no employees, there may be nowhere to store its location

Employee	Department	DepartmentLocation
234	Finance	Boston
223	Finance	Boston
399	Operations	Washington

38

Source: MIT 1.264 lecture notes

3rd Normal Form (2/2)

- **Solution**
  - Two entity types: Employee and department

Employee	Department
234	Finance
223	Finance
399	Operations

Department	DepartmentLocation
Finance	Boston
Operations	Washington

39

Source: MIT 1.264 lecture notes

Moral of Data Modeling Story

- Systems are ephemeral (short lived)
- Data is permanent
- If you mess up a system, you rewrite it and it's fixed
- If you mess up the data, it's usually irretrievable
- Real business have subtle business rules
  - Care in data modeling and business rules is needed to achieve good data quality
  - Care in data normalization is needed to preserve data quality

40

Source: MIT 1.264 lecture notes

Reference

Data Modeling

- http://ocw.mit.edu/
- http://phlonx.com/resources/nf3/
- www.agiledata.org/essays/dataModeling101.html
- http://db.grussell.org/section004.html

Free DB design tool:

- http://www.fabforce.net/dbdesigner4/

Normalization

- http://en.wikipedia.org/wiki/Database_normalization
- http://support.microsoft.com/kb/283878

Course Progress

- Software process
  - Requirement specification, Analysis, Design, Implementation, Testing & verification
- Rapid development: class mistake avoidance, applying fundamentals, risk management, schedule-oriented practices
- Virtualization → preparing testing environment
- Modeling
  - Software modeling – UML diagrams
  - Data modeling – ER diagram
- Data modeling → 1) clarify business rules, and 2) plan on how to store data in relational databases,

4th Normal Form (1/4)

Employee	Skill	Language
Brown	Cook	English
Smith	Type	German

A row should not contain two or more independent multi-valued facts about an entity.

- **Example that violates fourth normal form:**
  - An employee may have several skills and languages
- **Problems**
  - Uncertainty in how to maintain the rows. Several approaches are possible and different programmers may take different approaches, as shown on next slide

Source: MIT 1.264 lecture notes

4th Normal Form (2/4)
Problem 1

- **Disjoint format. Effectively same as 2 entity types.**
  - Blank fields ambiguous. Blank skill could mean:
    - Person has no skill
    - Attribute doesn't apply to this employee
    - Data is unknown
    - Data may be found in another record (as in this case)
  - Programmers will use all these assumptions over time, as will data entry staff and users

Employee	Skill	Language
Brown	Cook	
Brown	Type	
Brown		French
Brown		German
Brown		Greek

Source: MIT 1.264 lecture notes

4th Normal Form (3/4)
Problem 2

- **Cross product format**
  - Repetitions: updates must be done to multiple records and there can be inconsistencies
  - Insertion of a new skill may involve looking for a record with a blank skill, inserting a new record with possibly a blank language or skill, or inserting a new record pairing the skill with some or all of the languages.
  - Deletion is worse: It means blanking a skill in one or more records, and then checking you don't have 2 records with the same language and no skill, or it may mean deleting one or more records, making sure you don't delete the last mention of a language that should not be deleted

Employee	Skill	Language
Brown	Cook	French
Brown	Cook	German
Brown	Cook	Greek
Brown	Type	French
Brown	Type	German
Brown	Tye	Greek

Source: MIT 1.264 lecture notes

4th Normal Form (4/4)
Solution

- Two entity types: Employee-skill and employee-language
- Note that skills and languages may be related, in which case the starting example was ok:
  - If Smith can only cook French food, and can type in French and Greek, then skill and language are not multiple independent facts about the employee, and we have not violated fourth normal form.

Employee	Skill
Brown	Cook
Brown	Type

Employee	Language
Smith	French
Smith	German
Smith	Greek

Source: MIT 1.264 lecture notes

5th Normal Form (1/4)

Agent	Company	Product
Smith	Ford	Car
Smith	GM	Truck

A record cannot be reconstructed from several smaller record types.

- Example:
  - Agents represent companies
  - Companies make products
  - Agents sell products
- Most general case (allow any combination):
  - From the above table, however, Smith does not sell Ford trucks nor GM cars

Source: MIT 1.264 lecture notes

5th Normal Form (2/4)

- A variation is where the problem occurs–If an agent sells a certain product and she represents the company, then she sells that product for that company.

Agent	Company	Product
Smith	Ford	Car
Smith	Ford	Truck
Smith	GM	Car
Smith	GM	Truck
Jones	Ford	Car

- We can reconstruct all true facts from 3 tables instead of the 1:

Agent	Company
Smith	Ford
Smith	GM
Jones	Ford

Agent	Product
Smith	Car
Smith	Truck
Jones	Car

Company	Product
Ford	Car
Ford	Truck
GM	Car
GM	truck

Source: MIT 1.264 lecture notes

5th Normal Form (3/4)

- Problems with the 1 table form
  - Facts are recorded multiple times. E.g., the fact that Smith sells cars is recorded twice. If Smith stops selling cars, there are 2 rows to update and one will be missed.
  - Size of this table increases multiplicatively, while the normalized tables increase additively. With big operations, this is a big difference.
    - 100,000 x 100,000 is a lot bigger than 100,000 + 100,000

49

Source: MIT 1.264 lecture notes

5th Normal Form (4/4)

- An example with a subtle set of conditions

Agent	Company	Product
Smith	Ford	Car
Smith	Ford	Truck
Smith	GM	Car
Smith	GM	Truck
Jones	Ford	Car
Jones	Ford	Truck
Brown	Ford	Car
Brown	GM	Car
Brown	Toyota	Car
Brown	Toyota	Bus

Agent	Company
Smith	Ford
Smith	GM
Jones	Ford
Brown	Ford
Brown	GM
Brown	Toyota

Agent	Product
Smith	Car
Smith	Truck
Jones	Car
Jones	Truck
Brown	Car
Brown	Bus

Company	Product
Ford	Car
Ford	Truck
GM	Car
GM	Truck
Toyota	Car
Toyota	Bus

Jones sells cars and GM makes cars, but Jones does not represent GM
Brown represents Ford and Ford makes trucks, but Brown does notsell trucks
Brown represents Ford and Brown sells buses, but Ford does not make buses

50

Source: MIT 1.264 lecture notes