Review 2

CMPT 354

Jian Pei

jpei@cs.sfu.ca

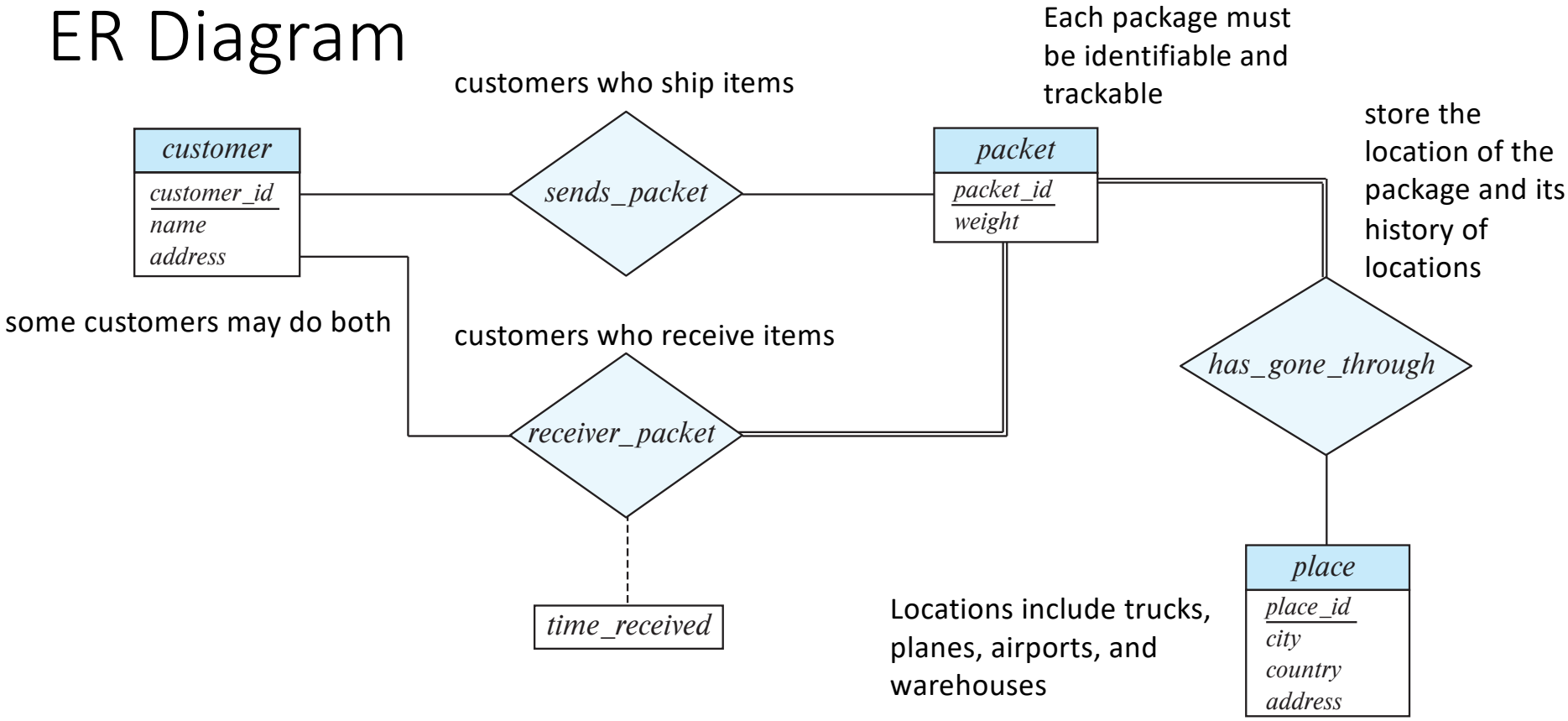
ER Design

- Overview of the Design Process
- The Entity-Relationship Model
- Complex Attributes
- Mapping Cardinalities
- Primary Key
- Removing Redundant Attributes in Entity Sets
- Reducing ER Diagrams to Relational Schemas
- Entity-Relationship Design Issues
- Aggregation
- **Specialization and generalization is NOT required in Midterm 2**

Example 1

- Design a database for a worldwide package delivery company (e.g., DHL or FedEx). The database must be able to keep track of customers who ship items and customers who receive items; some customers may do both. Each package must be identifiable and trackable, so the database must be able to store the location of the package and its history of locations. Locations include trucks, planes, airports, and warehouses.

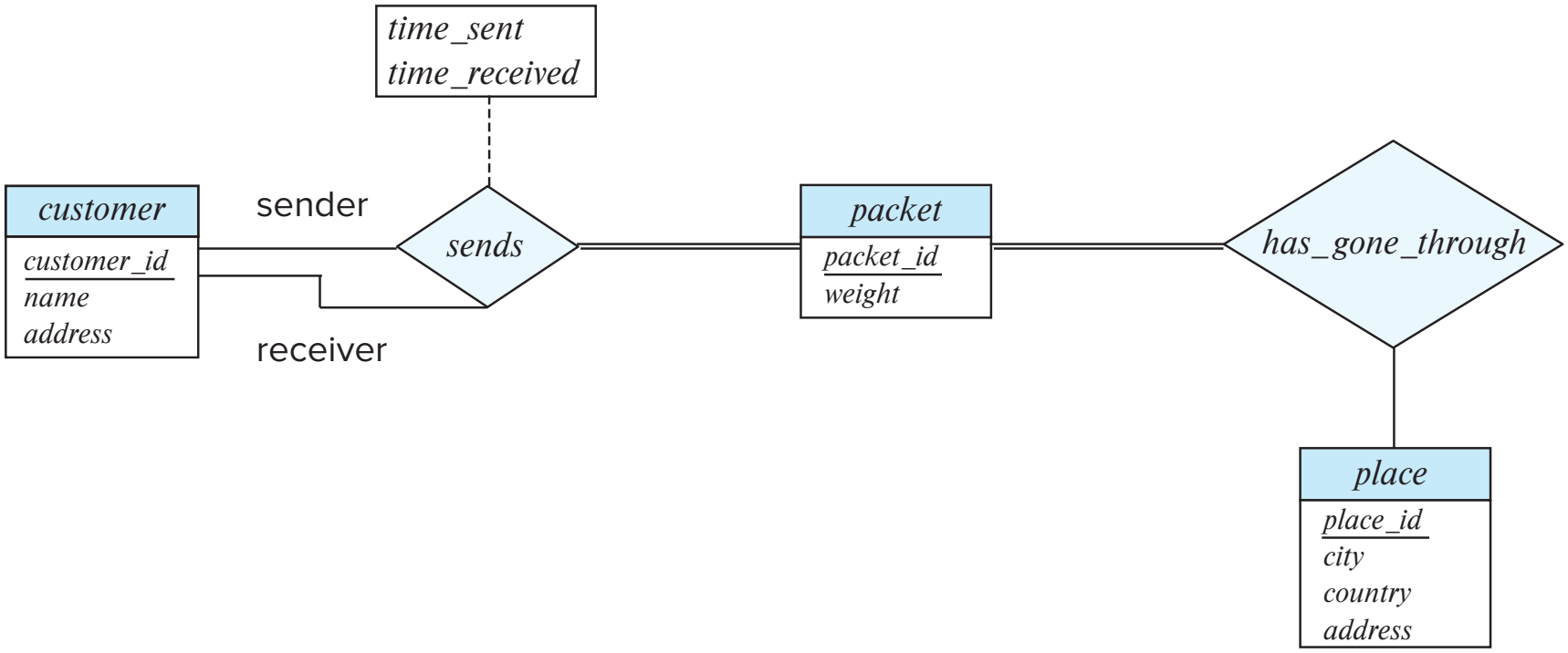
ER Diagram


customers who ship items
Each package must be identifiable and trackable
store the location of the package and its history of locations
customer
customer_id
name
address
sends_packet
packet
packet_id
weight
some customers may do both
customers who receive items
receiver_packet
has_gone_through
time_received
Locations include trucks, planes, airports, and warehouses
place
place_id
city
country
address

Schemas

customer(customer_id, customer_name, address)
package(package_id, weight, contents)
location(loc_id)
truck(loc_id, VIN)
plane(loc_id, type, mfg)
airport(loc_id, city, code)
warehouse(loc_id, address)
at(package_id, loc_id, time_in, time_out,
    foreign key *package_id* **references** *package,*
    foreign key *loc_id* **references** *location)*
receive(customer_id, package_id, time,
    foreign key *customer_id* **references** *customer,*
    foreign key *package_id* **references** *package)*
send(customer_id, package_id, time,
    foreign key *customer_id* **references** *customer,*
    foreign key *package_id* **references** *package)*

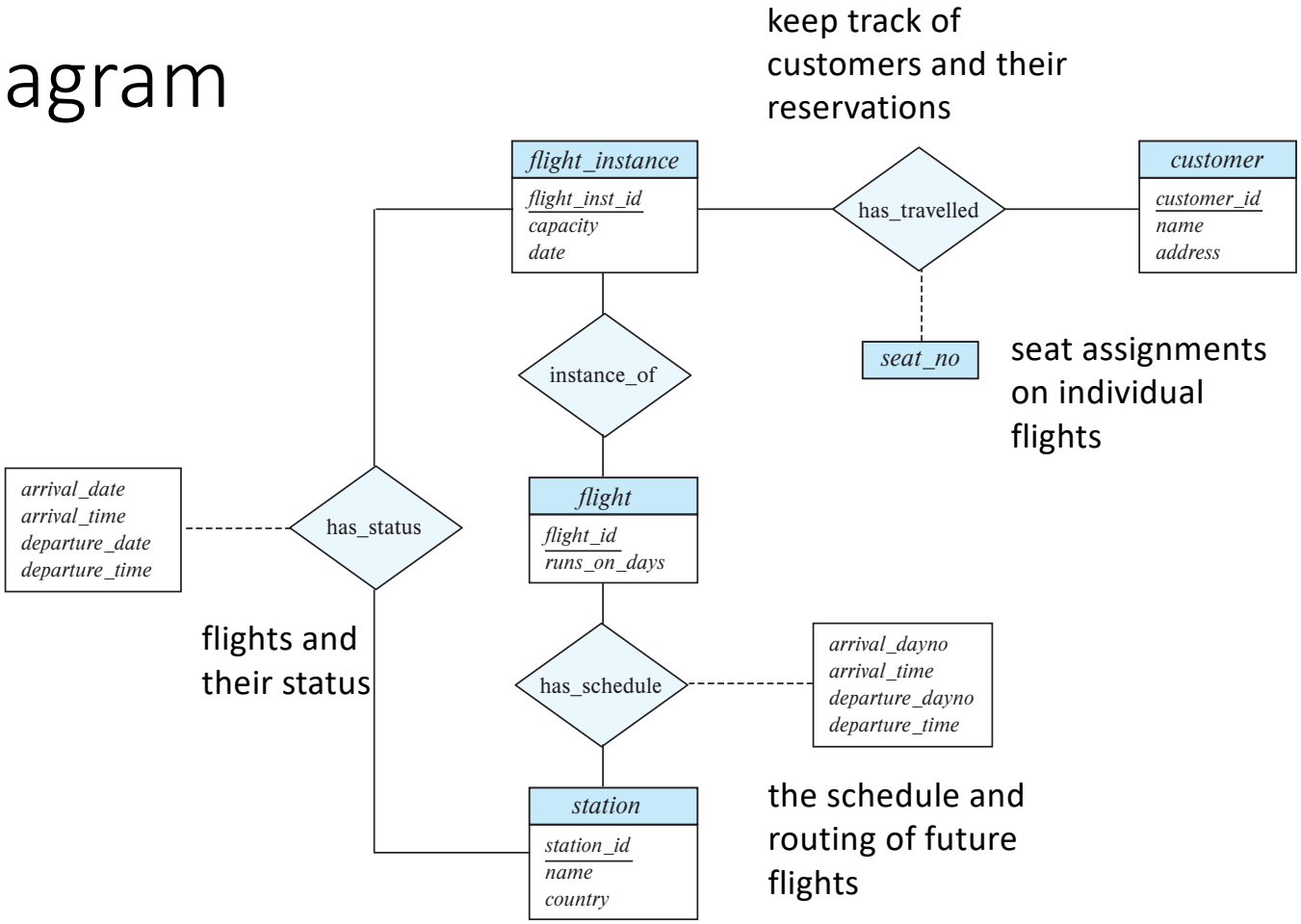
An Alternative Design



Example 2

- Design a database for an airline. The database must keep track of customers and their reservations, flights and their status, seat assignments on individual flights, and the schedule and routing of future flights.

ER Diagram


keep track of customers and their reservations
flight_instance
flight_inst_id
capacity
date
has_travelled
customer
customer_id
name
address
seat_no
seat assignments on individual flights
instance_of
arrival_date
arrival_time
departure_date
departure_time
has_status
flight
flight_id
runs_on_days
flights and their status
has_schedule
arrival_dayno
arrival_time
departure_dayno
departure_time
station
station_id
name
country
the schedule and routing of future flights

Schemas

flight_instance(flight_inst_id, capacity, date)
customer(customer_id, customer_name, address)
flight(flight_id, runs_on_days)
airport(airport_id, name, country)
has_traveled(flight_inst_id, customer_id, seat_number,
foreign key *flight_inst_id* **references** *flight_instance,*
foreign key *customer_id* **references** *customer)*
instance_of(flight_inst_id, flight_id,
foreign key *flight_inst_id* **references** *flight_instance,*
foreign key *flight_id* **references** *flight)*
has_schedule(flight_id, airport_id, arrival_time, departure_time,
foreign key *flight_id* **references** *flight,*
foreign key *airport_id* **references** *airport)*
has_status(flight_inst_id, airport_id, arrival_time, departure_time
foreign key *flight_inst_id* **references** *flight_instance,*
foreign key *airport_id* **references** *airport)*

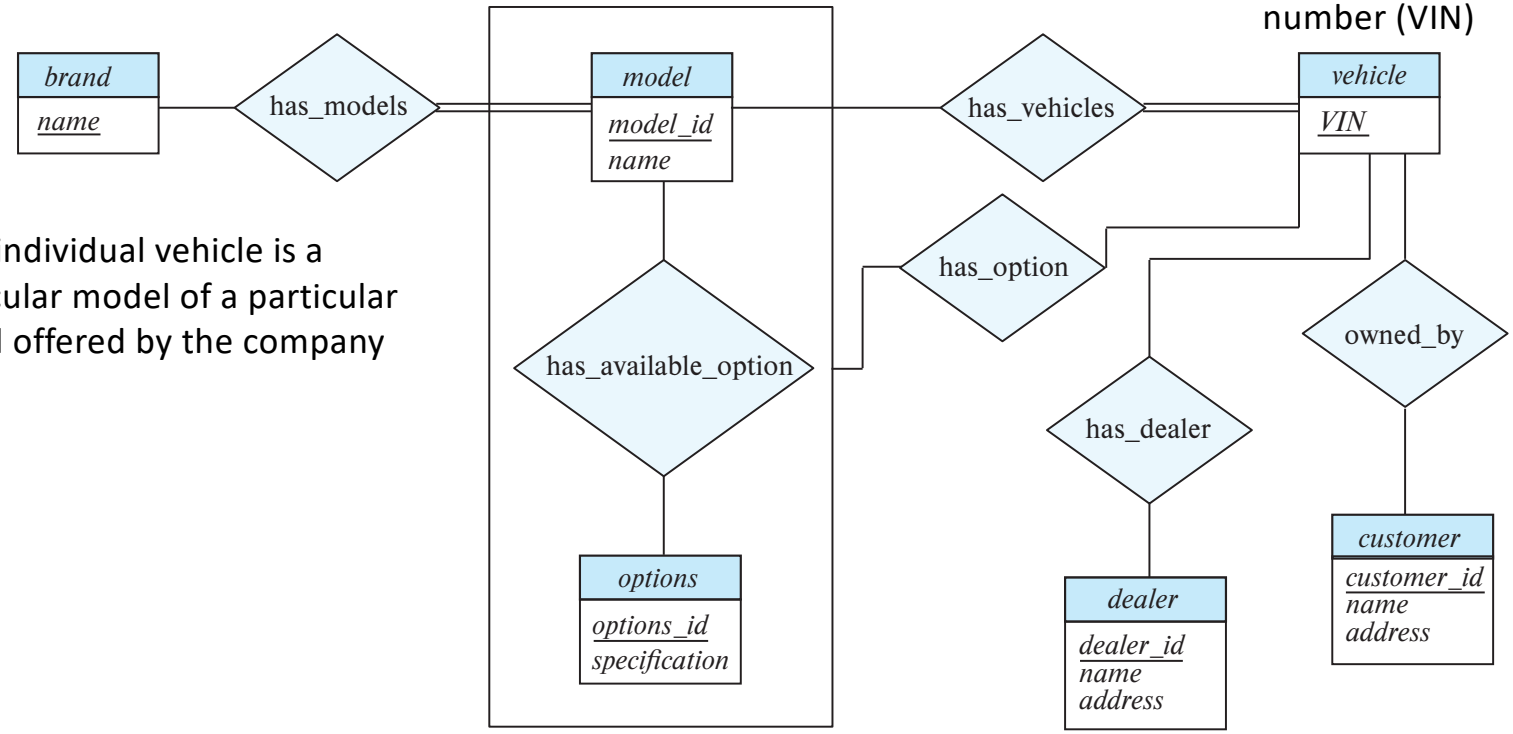
Example 3 (Aggregation)

- Design a database for an automobile company to provide to its dealers to assist them in maintaining customer records and dealer inventory and to assist sales staff in ordering cars.
- Each vehicle is identified by a vehicle identification number (VIN). Each individual vehicle is a particular model of a particular brand offered by the company (e.g., the XF is a model of the car brand Jaguar of Tata Motors). Each model can be offered with a variety of options, but an individual car may have only some (or none) of the available options. The database needs to store information about models, brands, and options, as well as information about individual dealers, customers, and cars.

ER Diagram

Each model can be offered with a variety of options, but an individual car may have only some (or none) of the available options

Each vehicle is identified by a vehicle identification number (VIN)



Each individual vehicle is a particular model of a particular brand offered by the company

store information about models, brands, and options, as well as information about individual dealers, customers, and cars

Schemas

brand(brand_name),
model(model_id, model_name)
vehicle(VIN, dealer_id, customer_id)
option(option_id, specification)
customer(customer_id, customer_name, address)
dealer(dealer_id, dealer_name, address)
has_model(brand_name, model_id ,
    foreign key *brand_name* **references** *brand,*
    foreign key *model_id* **references** *model)*
has_vehicle(model_id, VIN,
    foreign key *VIN* **references** *vehicle,*
    foreign key *model_id* **references** *model)*
has_available_option(model_id, option_id,
    foreign key *option_id* **references** *option,*
    foreign key *model_id* **references** *model)*
has_option(VIN, model_id, option_id,
    foreign key *VIN* **references** *vehicle,*
    foreign key *(model_id, option_id)* **references** *available_option)*
has_dealer(VIN, dealer_id ,
    foreign key *dealer_id* **references** *dealer,*
    foreign key *VIN* **references** *vehicle)*
owned_by(VIN, customer_id,
    foreign key *customer_id* **references** *customer,*
    foreign key *VIN* **references** *vehicle)*

Relational Design

- Features of Good Relational Design
- Functional Dependencies
- Decomposition Using Functional Dependencies
- Normal Forms
- Functional Dependency Theory
- Algorithms for Decomposition using Functional Dependencies

The Major Concepts

A relational schema R and a set of FDs F

Lossless and dependency preserving decomposition

Lossless decomposition, may be dependency preserving

3NF: for all $\alpha \rightarrow \beta$ in F^+ at least one of the following holds:

- $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \in \alpha$)
- α is a superkey for *R*
- Each attribute *A* in $\beta - \alpha$ is contained in a candidate key for *R*

BCNF: for all functional dependencies in F^+ of the form $\alpha \rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds: $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \subseteq \alpha$); or α is a superkey for *R*

Set of attributes α and closure α^+

Apply FDs

Canonical cover F_c ← Functional dependencies (FD) → FD closure F^+

A minimal set of FDs having the same closure as F^+

Complete set of FDs implied by F

About Midterm 2

- 8:30-9:20 am, 50 minutes
- SSCC 9001, not our regular classroom!
- Mask and ID
- No computer, cell phone, or calculator
- (Optional) a one-page, double sided cheat sheet
- Questions similar to those in Assignment 2
- How to prepare?
  - Textbook and lecture notes
  - To-Do lists
  - Assignment 2