LINKÖPING UNIVERSITY

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE (IDA)
DIVISION FOR DATABASE AND INFORMATION TECHNIQUES (ADIT)

Vinjetter Databasteknik

Vinjett 1: Local traffic in Miljököping (EER modeling)

to be worked on after we have covered Topic 4 (i.e., around week 47)

The municipal board of Miljököping wants to reduce car use and, instead, get more people to use public transport. A survey has been carried out in which two main reasons have been identified for citizens not traveling by public transport: First, it takes too long; that is, it is much faster to drive by car. Second, while it is admittedly cheaper than driving by car, it is cumbersome to buy tickets.

An alternative approach has therefore been decided:

- Public transport must be demand-driven. Persons who want to travel register themselves and specify the details of their desired travel (i.e., from where, to where, and when they want to go). This can be done some time in advance (for instance, every weekday I go to work at 8AM and home at 5PM), or with shorter notice. The person then receives a message about where the bus or car is going and exactly when.
- Booking of trips takes place via special booking companies. The booking can be made on the Web, by phone, or by SMS.
- The idea is to have many companies that can be responsible for the trips as well as for many different types of vehicles, from taxis to buses. Each company can see all desired travel bookings and choose to provide the transport. They can set it up as individual trips if needed or fixed routes for trips that many people travel.
- In order to achieve a sufficient mass of transports, it is expected that booking companies and transport companies must cooperate and be able to take over each other's bookings and transports.
- The people who travel pay a fixed cost per year. It is expected that this will be quite small and there is even hope to be able to offer the service free of charge at first.
- The companies responsible for the transports are paid per passenger they transport and the distance passengers must travel, where distance may be measured by the length of a direct line between the departure and the arrival points. The companies also get a bonus if they manage their transports efficiently; that is, the vehicle must be as full as possible and the total mileage must be minimal.
- The booking companies get paid according to a similar model. They receive a sum for each person who booked through them. In addition, there is a deduction for each passenger who does not get transported, has to wait too long for their transport, or ends up traveling longer than the double distance compared to taking their own car.

Can you draw an ER diagram for the above.

Vinjett 2: Relational database for road navigation

to be worked on after we have covered Topic 5 (i.e., around week 47-48 or later)

This vignette is about representing map and road information in a relational database. As a starting point, we may understand the road network as a graph similar to what it looked like in the optimization course. We assume that we can calculate a cost for each road section based on the distance and other available information (for example, information about traffic flows and weather). We want to use the information when searching for the most efficient route; that is, when applying one of the algorithms from the optimization course.

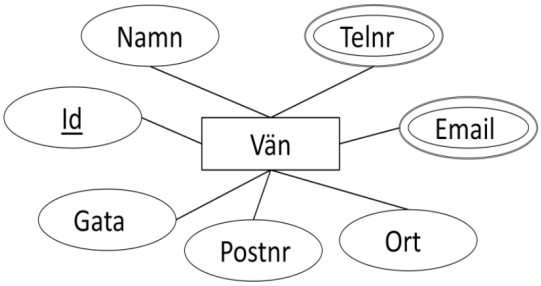
What would a good ER diagram for this information look like? What would a well-designed relational database look like? Is this affected by the algorithm's need for data? Start from the representation of graphs and the algorithms that you worked with in the optimization course. You are welcome to take a look at how maps are represented in GIS systems.

Vinjett 3: Translating EER schemas to relational database schemas

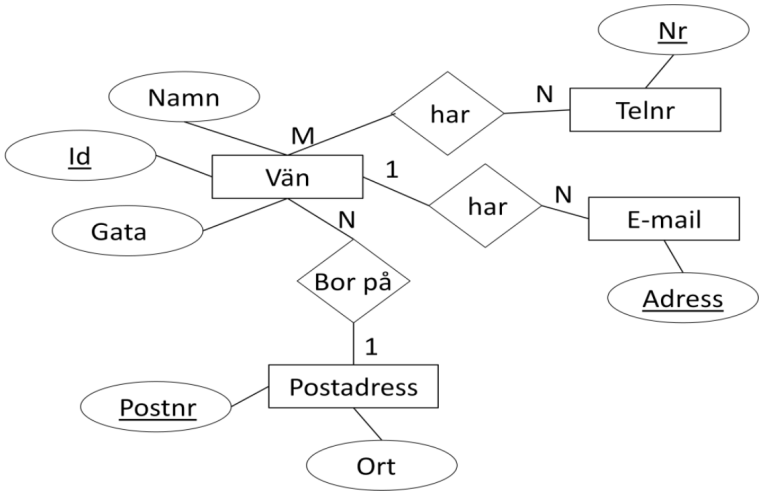
to be worked on after we have covered Topic 5 (i.e., around week 47-48 or later)

Suppose two people want to implement an address book application. Each of them proposes an EER diagram for a database to store the data of the address book (see below). How big are the differences between the two proposals? What will be the corresponding relational database schema?

Proposal 1:



Proposal 2:



Vinjett 4: Normalization, advantages and disadvantages

to be worked on after we have covered Topic 6 (i.e., around week 48-49 or later)

Consider the following three possible relational database schemas that capture information about which road segments a car is currently in.

Proposal 1:

tblRoadSegment
SegmentID (PK)
SegmentLength
….

tblCars
CarID (PK)
CurrentSegment (FK references tblRoadSegment(SegmentID))
CurrentSpeed
….

Proposal 2:

tblCarsInRoadSegment
SegmentID (PK tillsammans med CARID)
CarID (PK tillsammans med Segment ID)
CurrentSpeed
SegmentLength
….

Proposal 3:

tblRoadSegment
SegmentID (PK)
SegmentLength
NrOfCarsInSegment
….

tblCars
CarID (PK)
CurrentSegment (FK references tblRoadSegment(SegmentID))
CurrentSpeed
….

What is the difference between these proposals from a normalization perspective? Assume that a very important query for our database application is to keep track of how many cars are in each segment at any given moment. Expect that the database is large and that we have hundreds of thousands of active cars in the database. What are the respective disadvantages of the various representations (for instance, efficiency, easy to update, total amount of data, consistency)?

Two main types of database systems are usually expected: OLAP systems (online analytical processing), also called data warehouses, where users ask complex questions and data is rarely changed; and OLTP systems (online transaction processing) where it is expected that users update the database relatively frequently. Which of the above proposals is best suited for each of these systems?

How would you characterize the example application and which of the two types of systems is best here?

Vinjett 5: Transactions and actors

to be worked on after we have covered Topic 10 (i.e., around week 51)

The city of Bilköping has major problems with air pollution from cars. The solution will be to introduce tolls with the goal to limit and to balance the car flow. To this end, the idea is twofold: i) for each segment an upper bound on how many cars are allowed to pass per hour and ii) introduce a small fee for each segment of a road that a car drives on. The fee depends on how many cars are currently on the road segment. It is therefore cheaper to drive on roads with less traffic and more expensive to drive on segments where many currently want to drive. With this pricing model, it is hoped to be able to control car traffic so that road closures do not have to be done too often.

In order for this approach to work in practice, each car has been equipped with a special GPS module that can suggest a cheapest road calculated on the distance and the current price of the distances. The module can also pre-book tickets so that the drivers are guaranteed to be able to use the planned route. Booking of tickets and checking of where the cars drive takes place on top of a central database.

The following scenario describes the information flow between two cars and the server computer. For a trip, they first make a route planning based on the current traffic situation. Next, they reserve a place in all segments they need to drive on and, then, start driving. Each time they reach a new segment, this is reported by a call to *EnterSegment* which checks that the maximum upper bound for this hour has not been reached before a car can enter the segment. The following scenario arises (see next page):

Car 1:	**Car 2:**	
Start_transaction		
PlanRoute(Home, Work)		
ReserveSegment(A)		
	Start_transaction	
	PlanRoute(Home, City)	
	…reserve a number of segments	
ReserveSegment(B)		
	ReserveSegment(B)	
… reserve more segments		
… begin driving		
	… begin driving	
EnterSegment(A)		
EnterSegment(B)		
	EnterSegment (B)	Must wait if Car 1 has taken the last available spot
Commit		

Can this behavior be explained purely by transaction management? You can make assumptions about, for example, update strategy. Can you solve it with better transaction management? What happens if you change the update strategy? Is it possible to get a better solution if you change the meaning for *PlanRoute*, *ReserveSegment*, and *EnterSegment*?

For each solution: Is it realistic in practice in a situation with many cars? What can be done better?

Do you think that such a solution would improve the traffic situation in the city? What are the pros and cons?