

**Blekinge Tekniska
Högskola**
Institutionen för datavetenskap
ER-Modellering
Skapa ER-modell för en databas



Uppgiften är utfört av:

Ali Reza Nazari

Innehåll

Del 1: Konceptuell ER-Modellering.....	3
Del 1.1: Beskriv databasen I ett textstycke	3
Del 1.2: Skriv ned alla entiteter	4
Del 1.3: Skriv ned alla relationer och visa i en matris.....	5
Del 1.4: Rita enkelt ER-diagram med entiteter och relationer	6
Del 1.5: Komplettera ER-diagrammet med kardinaltal.....	7
Del 1.6: Komplettera ER-diagrammet med alla attribut sam kandidatnycklar	8
Del 2: Logisk Modellering	9
Del 2.1: Modifiera ER-diagrammet enligt relationsmodellen	9
Del 2.2: Utöka ER-diagrammet med primära och främmande nycklar samt kompletterande attribut	10
Del 3: Fysisk Modellering	12
Del 3.1: Skapa SQL DDL för tabellerna	12

Del 1: Konceptuell ER-Modellering

Del 1.1: Beskriv databasen I ett textstycke

Vi skall utveckla ett webbaserat system som hanterar online beställningar för e-handel företaget, BuckStar Online Shopping. Beställningen (order) innehåller kundinformation (från kundregistret), produktinformation (från produktregistret), lagerinformation (från lagerregistret) vidare kompletteras beställningen med en plocklista, faktura och händelselog.

Beställningen innehåller kundinformation (kontaktdetaljer) som i sin tur är lagrad i kundregistret.

Beställningen består av en eller flera produkter som i sin tur är indelat i olika kategorier och för varje produkt finns det produktinformation lagrad (produkter med produktkod, namn, kort beskrivning och pris).

Beställningen har en plocklista som ärver från ordern. Vidare innehåller den plockinformation(lagerhylla)

Beställningen har en faktura som ärver från ordern och den innehåller också prisinformation (pris per produktrad och det summerade priset)

Beställningen har en händelselog som innehåller viktiga händelser i systemet (när order skapades, faktura skapades, produkterna plockades)

Del 1.2: Skriv ned alla entiteter

Vi skall utveckla ett webbaserat system som hanterar online beställningar för e-handel företaget, BuckStar Online Shopping. Beställningen (**order**) innehåller kundinformation (från kundregistret), produktinformation (från produktregistret), lagerinformation (från lagerregistret) vidare kompletteras beställningen med en plocklista, faktura och händelselog.

Beställningen innehåller en **kund** (kontaktdetaljer) som i sin tur lagrad information i **kundregistret**.

Beställningen består av en eller flera **produkter** som i sin tur är indelat i olika **kategorier** och for varje produkt finns det produktinformation lagrad (produkter med produktkod, namn, kort beskrivning och pris).

Beställningen har en **plocklista** som ärver från ordern. Vidare innehåller det plockinformation(**lagerhylla**)

Beställningen har en **faktura** som ärver från ordern och den innehåller också prisinformation (pris per produktrad och det summerade priset)

Beställningen har en **händelselog** som innehåller viktiga hädelser i systemet (när order skapades, faktura skapades, produkterna plockades)

Följande entiteter finns i systemet:

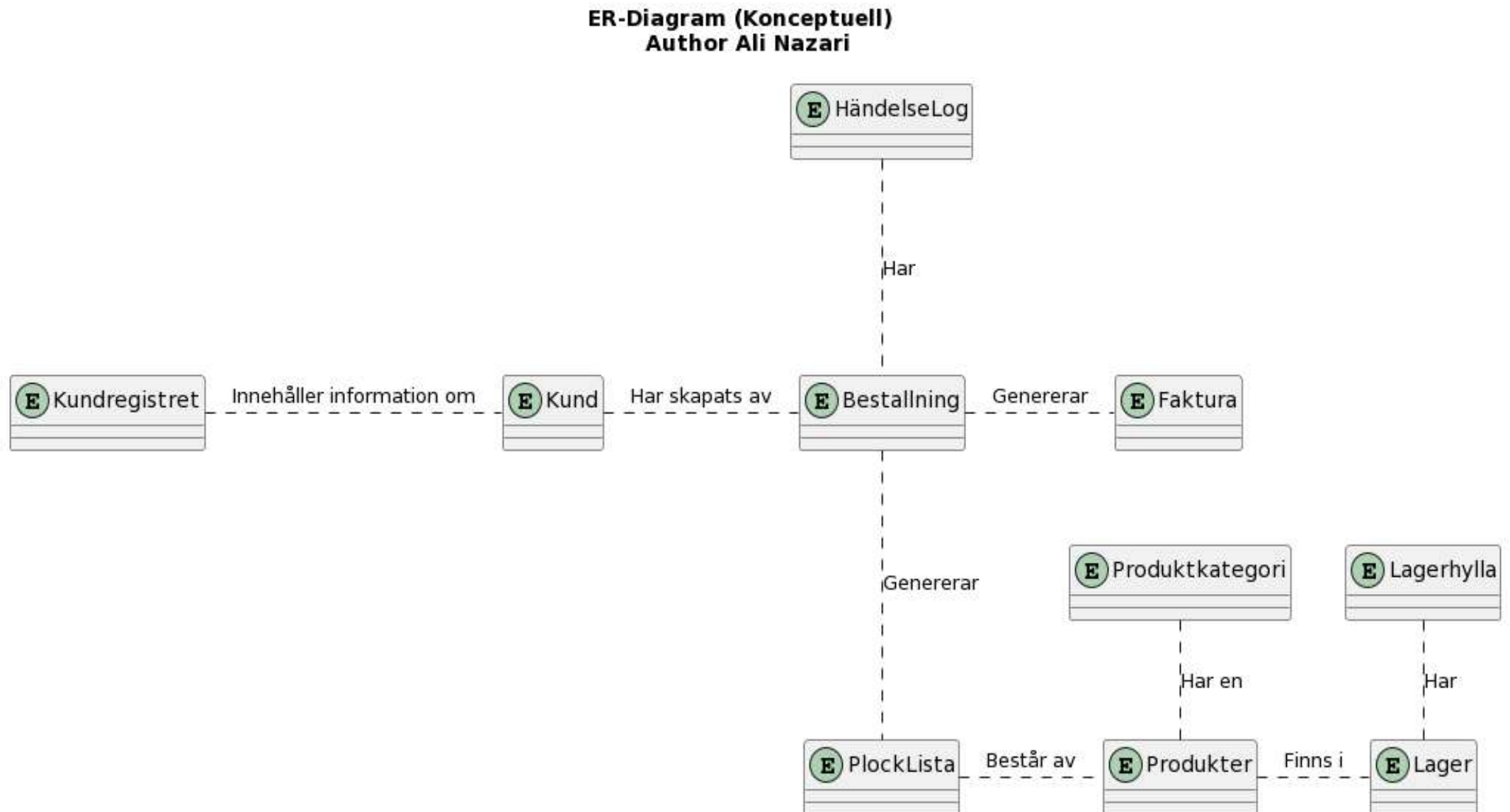
- Order
- Kund
- Kundregistret
- Lager
- Produkter
- Kategori
- Plocklista
- Faktura
- Händelselog

Del 1.3: Skriv ned alla relationer och visa i en matris

Entiteter	Beställning	Kund	Kundregister	Produkter	Lager	LagerHylla	Plocklista	Faktura	Händelselog	Kategori
Beställning		Har skapats av		Består av			Genererar	Genererar	Har	
Kund	Har		Innehåller information om							
Kundregister		Innehåller information om								
Produkter					Finns i					Tillhör till en
Lager				Innehåller	Finns i	Har många				
LagerHylla										
Plocklista				Består av	Visar antal					
Faktura		Innehåller		Innehåller						
Händelselog	Sparar	Sparar						Sparar		
Kategori	Består av en eller flera produkter									

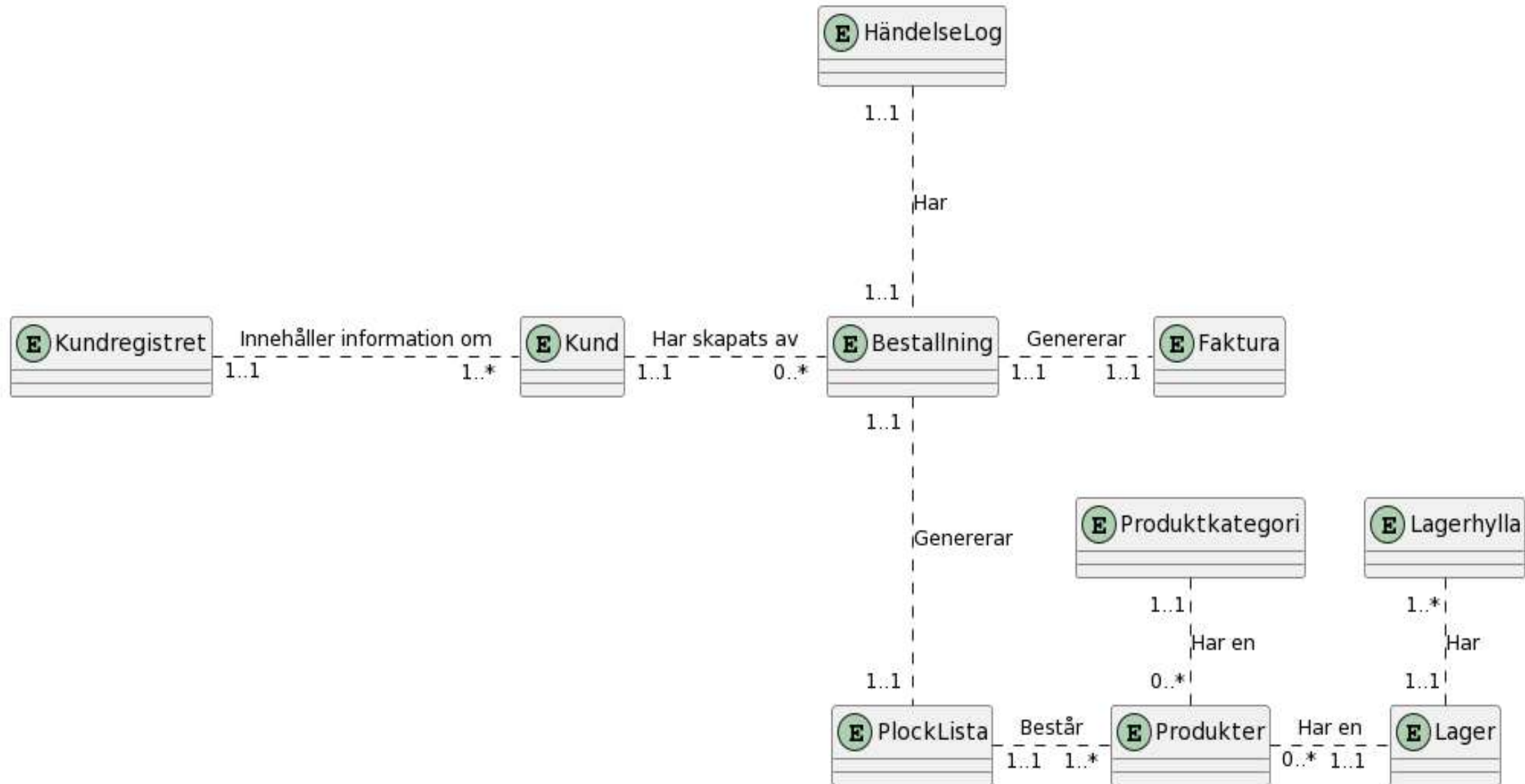
- Beställning har skapats av en kund, består av en eller flera produkter, generar en plocklista, generar en faktura och har ett händelselog.
- Kund hämtas från kundregistret.
- Kundregister innehåller information om kunden
- Produkter finns i lager.
- Produkter tillhör en kategori
- Ett lager består av många lagerhyllor
- Plocklista består av en eller flera produkter som finns i lagerhylla.
- Faktura innehåller både information om kunden och produktinformation.
- Händelselog sparar ordern, kundinformationen och fakturan.

Del 1.4: Rita enkelt ER-diagram med entiteter och relationer

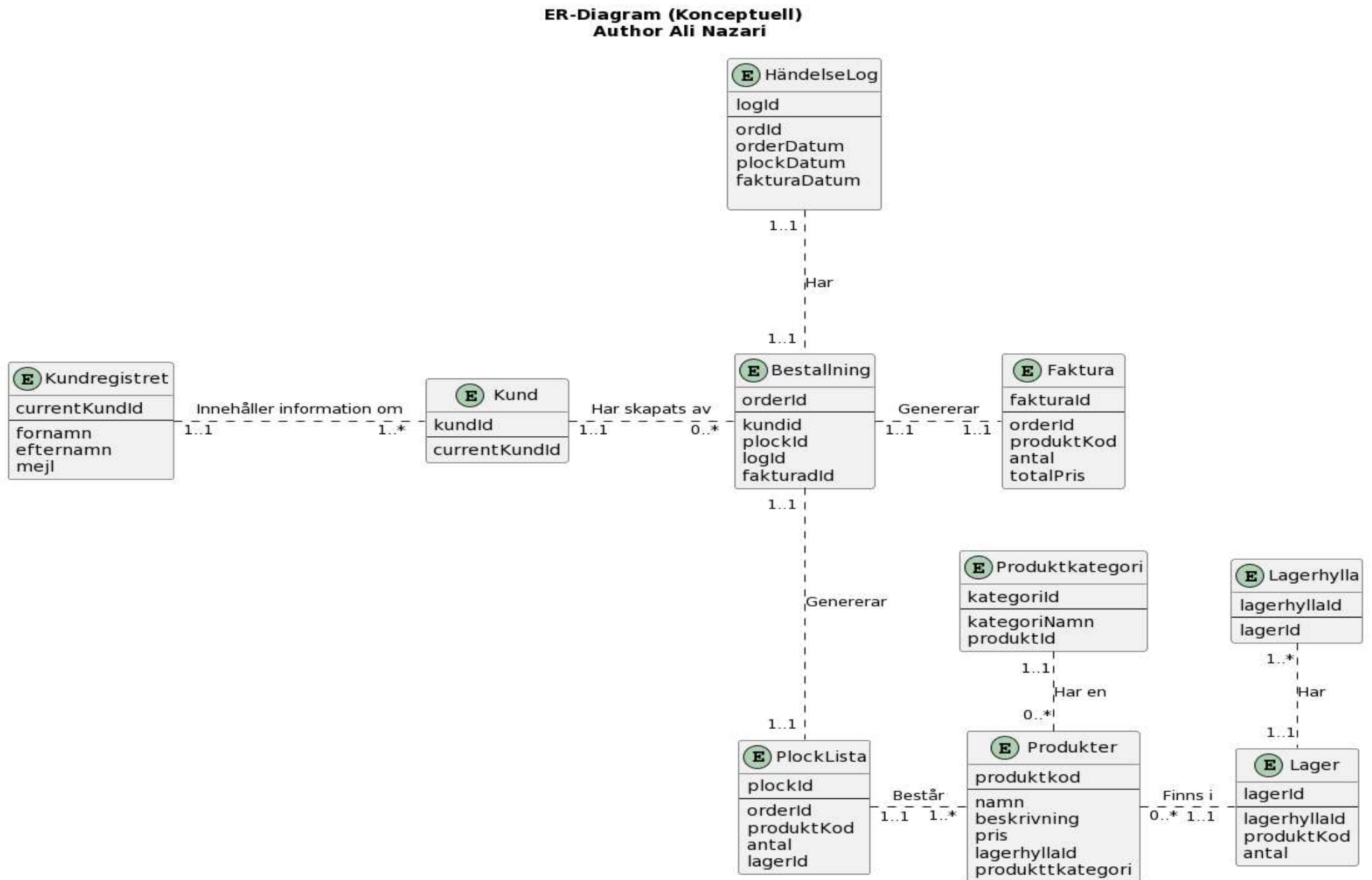


Del 1.5: Komplettera ER-diagrammet med kardinaltal

ER-Diagram (Konceptuell)
Author Ali Nazari



Del 1.6: Komplettera ER-diagrammet med alla attribut sam kandidatnycklar



Del 2: Logisk Modellering

Del 2.1: Modifiera ER-diagrammet enligt relationsmodellen

Diagrammet är uppdaterat enligt relationsmodellen, dvs följande finns inte med i diagrammet:

- Inga många-många förhållanden (N:M).
- Inga flervärdesattribut.
- Endast ett värde i varje cell.
- Varje rad är unik.
- Inga komplexa relationer.

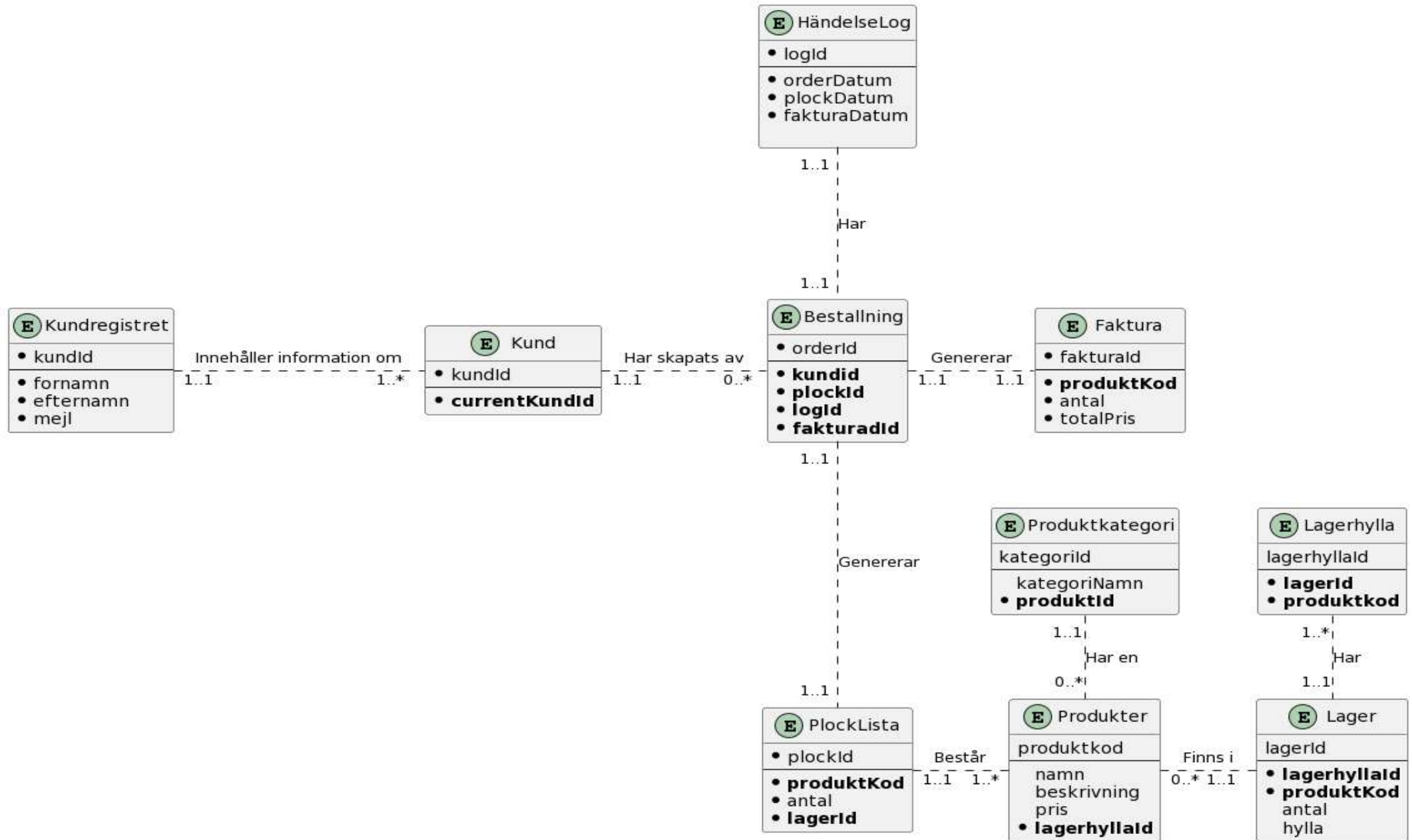
Vidare ser diagrammet samma som i del 1.6 på sida 8.

Nedan är en text-mässig beskrivning av det uppdaterade ER-diagrammet:

- Beställning (orderId, kundId, plockId, logId, fakturaId)
- Kund (kundId, currentkundId)
- Kundregistret (kundId, fornamn, efternamn, mejl)
- HändelseLog (logId, orderDatum, plockDatum, fakturaDatum)
- Faktura (fakturaId, produktkod, antal, totalPris)
- PlockLista (plockId, orderId, produktkod, antal, lagerId)
- Produkter (produktkod, namn, beskrivning, pris, kategoriId, lagerhyllaId)
- Produktkategori (kategoriId, kategoriNamn, produktId)
- Lager (lagerId, lagerhyllaId, produktKod, antal)
- Lagerhylla (lagerhyllaId, lagerId, produktkod)

Del 2.2: Utöka ER-diagrammet med primära och främmande nycklar samt kompletterande attribut

ER-Diagram (Konceptuell)
Author Ali Nazari



Nedan är en text-mässig beskrivning av det uppdaterade ER-diagrammet:

- Beställning (orderId, #kundId, #plockId, #logId, #fakturaId)
- Kund (#kundId, #orderId)
- Kundregistret (kundId, fornamn, efternamn, mejl)
- HändelseLog (logId, #orderId, orderDatume, plockDatum, fakturaDatum)
- Faktura (fakturaId, #orderId, #produktId, antal, totalPris)
- PlockLista (plockId, #orderId, #produktKod, antal, #lagerId)
- Produkter (produktkod, namn, beskrivning, pris)
- Produktkategori (kategoriId, kategoriNamn, #produktId)
- Lager (lagerId, #produktKod, antal, hylla)
- Lagerhylla (lagerhyllaId, #lagerId, #produktkod)

Del 3: Fysisk Modellering

Del 3.1: Skapa SQL DDL för tabellerna

Jag har skrivit DDL för databasen per hand. Efter att ha försökt att hitta hur man lägger till DDL som ett appendix kunde jag tyvärr lyckas så jag kommer att klistra det in som text. Vidare kommer ddl.sql och setup.sql finnas i mappen ers2. Jag hade lite svårt att skap SQL DDL för det var många tabeller som gick in i varandra, dvs med främmande nycklar. Men jag försökte mig fram genom att droppa de child-tabellerna först.

```
--
```

```
-- DDL for database eshop
```

```
--
```

```
USE eshop;
```

```
DROP TABLE IF EXISTS Bestallning;
```

```
DROP TABLE IF EXISTS Kund;
```

```
DROP TABLE IF EXISTS Kundregistret;
```

```
DROP TABLE IF EXISTS PlockLista;
```

```
DROP TABLE IF EXISTS Lagerhylla;
```

```
DROP TABLE IF EXISTS Lager;
```

```
DROP TABLE IF EXISTS Faktura;
```

```
DROP TABLE IF EXISTS Produktkategori;
```

```
DROP TABLE IF EXISTS Produkter;
```

```
DROP TABLE IF EXISTS HandelseLog;
```

```
CREATE TABLE Kundregistret (  
    kundId INT AUTO_INCREMENT,  
    fornamn VARCHAR(20),  
    efternamn VARCHAR(20),  
    mejl VARCHAR(20),  
    PRIMARY KEY (kundId)  
);
```

```
CREATE TABLE Produkter (  
    produktkod INT AUTO_INCREMENT,  
    namn VARCHAR(20),  
    beskrivning VARCHAR (200),  
    pris INT,  
    PRIMARY KEY (produktkod)  
);
```

```
CREATE TABLE Produktkategori (  
    kategoriId INT AUTO_INCREMENT,  
    kategoriNamn VARCHAR(50),  
    produktkod INT,  
    PRIMARY KEY (kategoriId),  
    FOREIGN KEY (produktkod) REFERENCES Produkter(produktkod)  
);
```

```
CREATE TABLE HandelseLog (  
    logId INT AUTO_INCREMENT,  
    orderDatum DATE,  
    plockDatum DATE,  
    fakturaDatum DATE,  
    PRIMARY KEY (logId)  
);
```

```
CREATE TABLE Lager (  
    lagerId INT AUTO_INCREMENT,  
    produktkod INT,  
    antal INT,
```

```
hylla VARCHAR(20),  
PRIMARY KEY (lagerId),  
FOREIGN KEY (produktkod) REFERENCES Produkter(produktkod)  
);
```

```
CREATE TABLE Lagerhylla (  
    lagerhyllaId INT AUTO_INCREMENT,  
    lagerId INT,  
    produktkod INT,  
    PRIMARY KEY (lagerhyllaId),  
    FOREIGN KEY (lagerId) REFERENCES Lager(lagerId),  
    FOREIGN KEY (produktkod) REFERENCES Produkter(produktkod)  
);
```

```
CREATE TABLE Kund (  
    kundId INT AUTO_INCREMENT,  
    PRIMARY KEY (kundId),  
    FOREIGN KEY (kundId) REFERENCES Kundregistret(kundId)  
);
```

```
CREATE TABLE Faktura(  
    fakturaId INT AUTO_INCREMENT,
```

```
        fakturaId INT AUTO_INCREMENT,  
    produktKod INT,  
    antal INT,  
    totalPris INT,  
    PRIMARY KEY (fakturaId),  
    FOREIGN KEY (produktkod) REFERENCES Produkter(produktkod)  
);
```

```
CREATE TABLE PlockLista (  
        plockId INT AUTO_INCREMENT,  
    produktKod INT,  
    antal INT,  
    lagerId INT,  
    PRIMARY KEY (plockId),  
    FOREIGN KEY (produktkod) REFERENCES Produkter(produktkod)  
);
```

```
CREATE TABLE Bestallning (  
        orderId INT AUTO_INCREMENT,
```



```
kundId INT,  
plockId INT,  
logId INT,  
fakturaId INT,  
PRIMARY KEY (orderId),  
FOREIGN KEY (kundId) REFERENCES Kundregistret(kundId),  
FOREIGN KEY (plockId) REFERENCES PlockLista(plockId),  
FOREIGN KEY (logId) REFERENCES HandelseLog(logId),  
FOREIGN KEY (fakturaId) REFERENCES Faktura(fakturaId)  
);
```