ER-MODELL AV EN E-BUTIKS DATABAS

Krystyna Kalinchuk
krka21@student.bth.se

Contents

Steg 1 – Konceptuell modellering

Databasens beskrivning

“ Databasen behöver hantera ett kundregister (kunder med kontaktdetaljer), ett produktregister (produkter med produktkod, namn, kort beskrivning och pris) där varje produkt finns i en eller flera produktkategorier.

Databasen behöver också innehålla ett lager där man ser hur många av varje produkt som finns i lagret och en notering om var produkten ligger i lagret (vilken hylla). En och samma produkt kan vara utspridd över olika hyllor i lagret.

När kunden beställer en produkt så skapas en order som innehåller kundens detaljer tillsammans med vilka produkter som beställts och dess beställda antal.

Utifrån ordern skapas en plocklista som kan skickas till lagret för leverans. Plocklistan innehåller samma information som ordern, men med tillägget att varje produktrad mappas mot en lagerhylla så att lagerpersonalen kan se vilken hylla de kan hämta produkten på.

När leveransen är packad så bifogas en faktura som har samma innehåll som ordern men nu med priset per produktrad och det summerade priset.

Det skall finnas en logg där man kan se viktiga händelser i systemet, vad hände, när hände det. Det kan till exempel vara när order/faktura skapades eller raderades.”

Identifiera entiteter

“ Databasen behöver hantera **ett kundregister** (kunder med kontaktdetaljer), **ett produktregister** (produkter med produktkod, namn, kort beskrivning och pris) där varje produkt finns i en eller flera produktkategorier.

Databasen behöver också innehålla **ett lager** där man ser hur många av varje produkt som finns i lagret och en notering om var produkten ligger i lagret (vilken hylla). En och samma produkt kan vara utspridd över olika hyllor i lagret.

När kunden beställer en produkt så skapas **en order** som innehåller kundens detaljer tillsammans med vilka produkter som beställts och dess beställda antal.

Utifrån ordern skapas **en plocklista** som kan skickas till lagret för leverans. Plocklistan innehåller samma information som ordern, men med tillägget att varje produktrad mappas mot en lagerhylla så att lagerpersonalen kan se vilken hylla de kan hämta produkten på.

När leveransen är packad så bifogas **en faktura** som har samma innehåll som ordern men nu med priset per produktrad och det summerade priset.

Det skall finnas en logg där man kan se viktiga händelser i systemet, vad hände, när hände det. Det kan till exempel vara när order/faktura skapades eller raderades.”

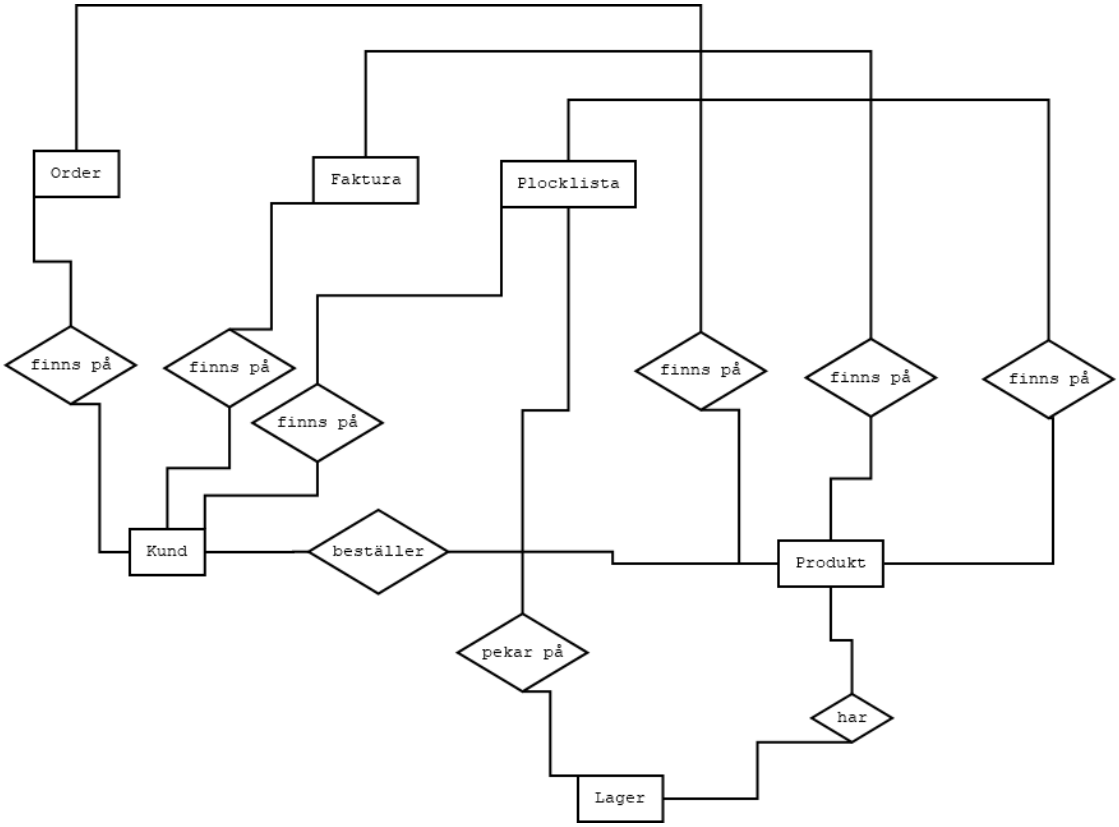
Hittade entiteter:

- Kundregister (kunden, kontaktdetaljer)
- Produktregister (produktkod, namn, kort beskrivning, pris, produktkategori)
- Lager (hur många av produkt finns, på vilken hylla den ligger)
- Order (kunden, produkt, antal)
- Plocklista (kunden, produkt, antal, lagerhylla)
- Faktura (kunden, produkt, antal, pris per produkt, det totala summan)

Relationer i en matris

Entiteter	***Kund***	***Produkt***	***Lager***	***Order***	***Plocklista***	***Faktura***
Kund		beställer		finns på	finns på	finns på
Produkt	beställs av		ligger på	finns på	finns på	finns på
Lager		har				
Order	har	har				
Plocklista	har	har	pekar på			
Faktura	har	har				

ER-diagram med entiteter och relationer



Kund beställer produkt

Kund finns på order

Kund finns på plocklista

Kund finns på faktura

Produkt kategoriseras enligt produktkategori

Produkt ligger på lager

Produkt finns på order

Produkt finns på plocklista

Produkt finns på faktura

Lager har produkt

Plocklista pekar på lager

ER-diagram med kardinalitet

M:N kund kan beställa en eller flera produkter, produkt kan beställas av en eller flera kunder

1:N order har en kund, kund kan finnas på flera ordrar

1:N plocklista har en kund, kund kan finnas på flera plocklistor

1:N faktura har en kund, kund kan finnas på flera fakturor

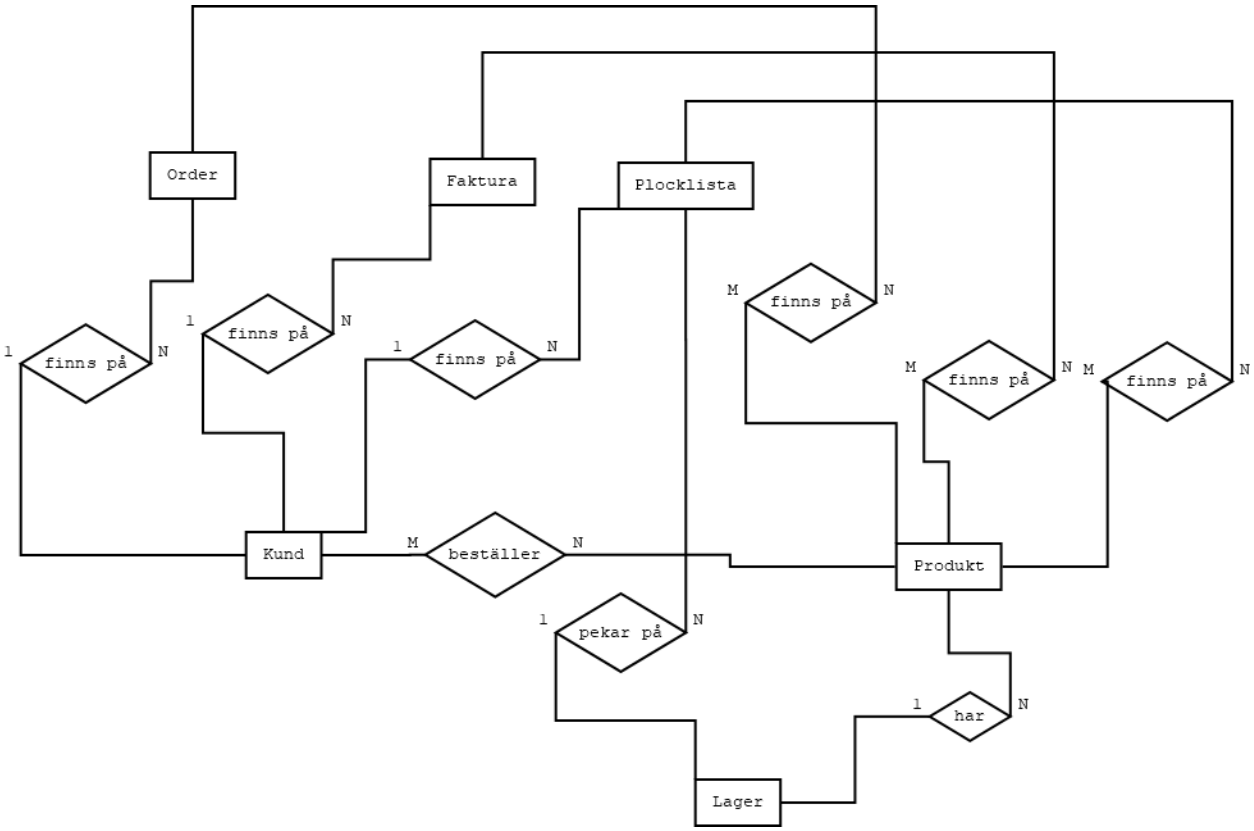
M:N produkt finns på en eller flera ordrar, order kan innehålla flera produkter

M:N produkt finns på flera plocklistor, plocklista kan innehålla flera produkter

M:N produkt finns på en eller flera fakturor, faktura kan innehålla flera produkter

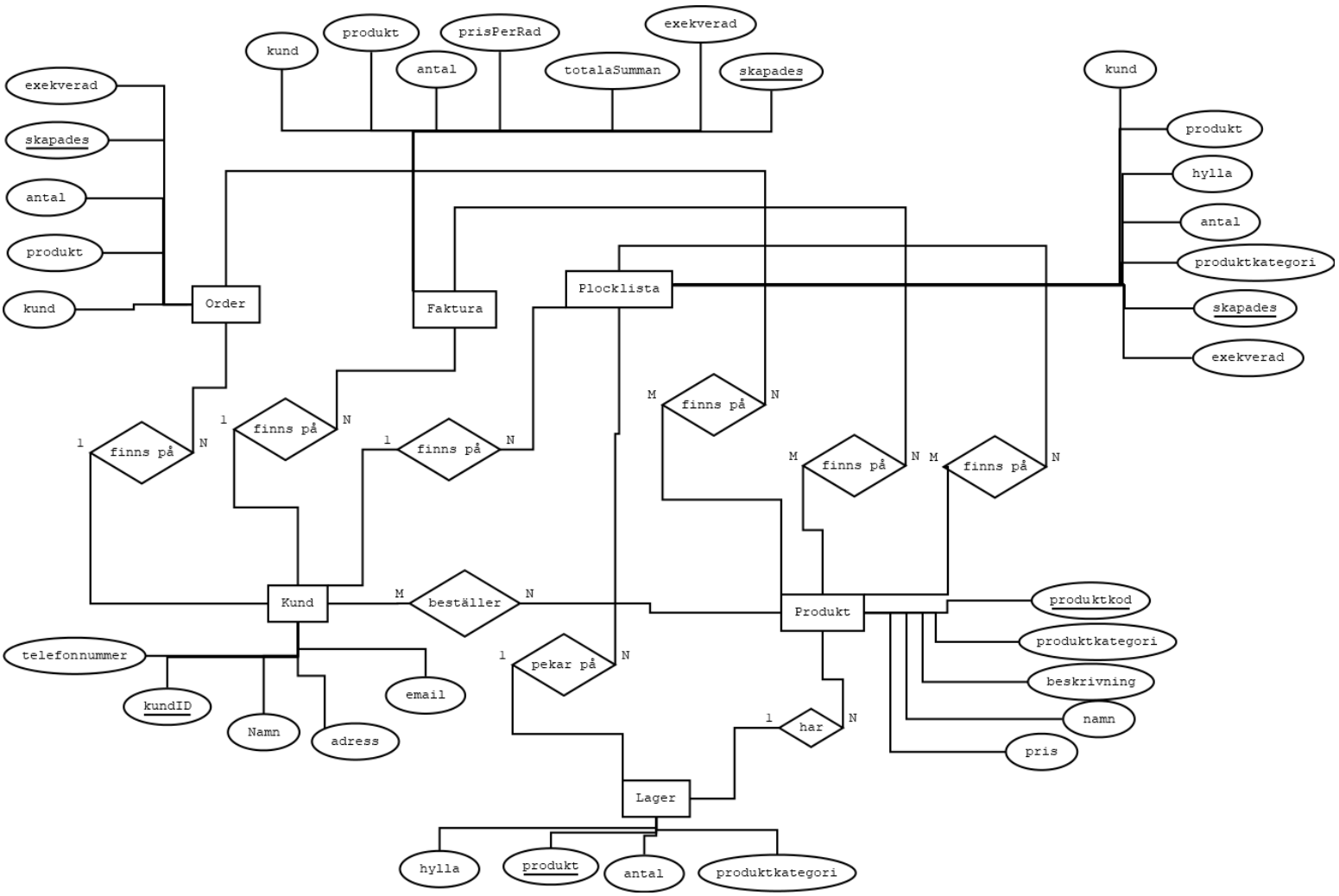
1:N plocklista pekar på ett lager, lager kan ha flera plocklistor som pekar på det

1:N produkt finns på ett lager, lager har flera produkter



ER-diagram med attribut och kandidatnycklar

- Kundregister (*kundID*, namn, adress, telefonnummer, email)
- Produktregister (*produktkod*, namn, beskrivning, pris, produktkategori)
- Lager (*produkt*, antal, produktkategori, hylla)
- Order (kund, produkt, antal, *skapades*, exekverades)
- Plocklista (kund, produkt, antal, produktkategori, lagerhylla, *skapades*, exekverades)
- Faktura (kund, produkt, antal, pris per rad, totala summan, *skapades*, exekverades)

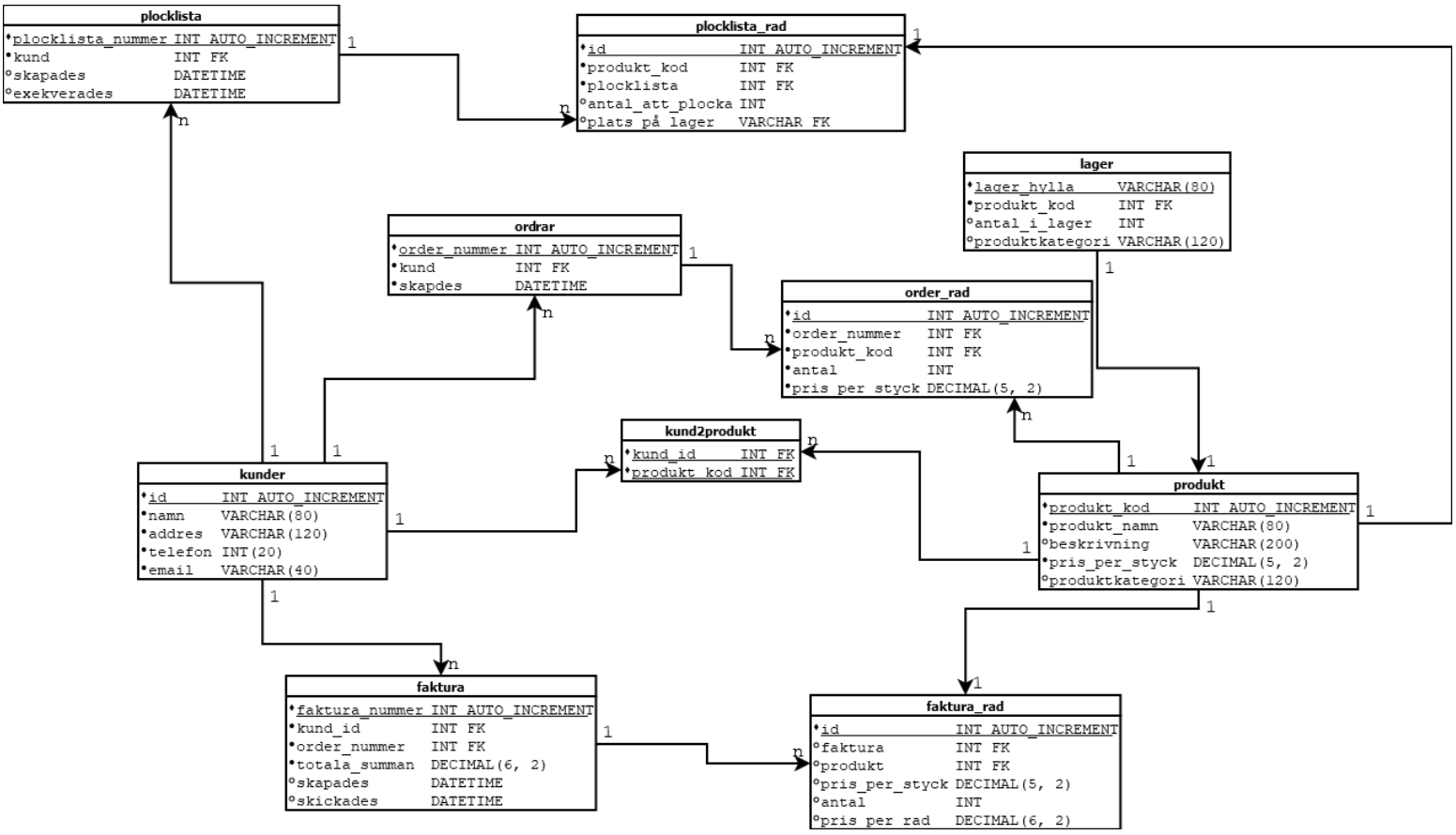


Steg 2 – Logisk modellering

Nedan är listade tabeller med attribut som kommer att finnas i databas ”Eshop”.

Primära nycklar är i fet typsnittstil, främmande nycklar är kursiva.

- Kunder (**id,** namn, adress, telefon, email)
- Produkt (**produkt_kod,** produkt_namn, beskrivning, pris_per_styck, produktkategori)
- Kund2produkt(**kund, produkt_kod**)
- Lager (**lager_hylla,** *produkt_kod*, antal_i_lager, produktkategori)
- Ordrar (**order_nummer,** *kund*, skapades, skickades)
- Order_rad(**id**, *order_nummer*, *produkt_kod*, antal, pris_per_styck)
- Plocklista (**plocklista_nummer,** *kund*, skapades, exekverades)
- Plocklista_rad (**id**, *produkt_kod*, *plocklista*, antal_att_plocka, plats_på_lager)
- Faktura (**faktura_nummer**, *kund_id*, *order_nummer*, totala_summan, skapades, skickades)
- Faktura_rad (**id**, *faktura_nummer*, *produkt_id*, pris_per_styck, antal, pris_per_rad)



Steg 3 – Fysisk modellering

SQL DDL kod för databasen “Eshop” har jag skrivit själv och testkörde i Workbench, allt verkar fungera bra. Just nu så finns det 10 tabeller i databasen med antingen ett-till-många, många-till-ett eller ett-till-ett relationer.

Appendix

```
DROP DATABASE IF EXISTS eshop;
CREATE DATABASE eshop;
USE eshop;
```

SQL DDL:

```
DROP TABLE IF EXISTS order_rad;
DROP TABLE IF EXISTS plocklista_rad;
DROP TABLE IF EXISTS plocklista;
DROP TABLE IF EXISTS lager;
DROP TABLE IF EXISTS kund2produkt;
DROP TABLE IF EXISTS faktura_rad;
DROP TABLE IF EXISTS faktura;
DROP TABLE IF EXISTS produkt;
DROP TABLE IF EXISTS ordrar;
DROP TABLE IF EXISTS kunder;

CREATE TABLE kunder (
   id INT AUTO_INCREMENT,
   namn VARCHAR(80),
   adress VARCHAR(120),
   telefon INT,
   email VARCHAR(40),

   PRIMARY KEY (id)
);
```

```
CREATE TABLE  produkt (
   produkt_kod INT AUTO_INCREMENT,
   produkt_namn VARCHAR(80),
   beskrivning VARCHAR(200),
   pris_per_styck DECIMAL(5, 2),
   produktkategori VARCHAR(120),

   PRIMARY KEY (produkt_kod)
);

CREATE TABLE kund2produkt(
   kund INT,
   produkt_kod INT,

   PRIMARY KEY (kund, produkt_kod),
   FOREIGN KEY (kund) REFERENCES kunder(id),
   FOREIGN KEY (produkt_kod) REFERENCES produkt(produkt_kod)
);

CREATE TABLE lager(
   lager_hylla VARCHAR(80),
   produkt_kod INT,
   antal_i_lager INT,
   produktkategori VARCHAR(120),

   PRIMARY KEY (lager_hylla),
   FOREIGN KEY (produkt_kod) REFERENCES produkt(produkt_kod)
);
```

```
CREATE TABLE ordrar(
   order_nummer INT AUTO_INCREMENT,
   kund INT,
   skapades DATETIME DEFAULT CURRENT_TIMESTAMP,
   skickades DATETIME DEFAULT NULL,

   PRIMARY KEY (order_nummer),
   FOREIGN KEY (kund) REFERENCES kunder(id)
);

CREATE TABLE order_rad(
   id INT,
   order_nummer INT,
   produkt_kod INT,
   antal INT,
   pris_per_styck DECIMAL(5, 2),

   PRIMARY KEY (id),
   FOREIGN KEY (order_nummer) REFERENCES ordrar(order_nummer),
   FOREIGN KEY (produkt_kod) REFERENCES produkt(produkt_kod)
);
```

```
CREATE TABLE plocklista(
   plocklista_nummer INT AUTO_INCREMENT,
   kund INT,
   skapades DATETIME DEFAULT CURRENT_TIMESTAMP,
   exekverades DATETIME DEFAULT NULL,

   PRIMARY KEY (plocklista_nummer),
   FOREIGN KEY (kund) REFERENCES kunder(id)
);

CREATE TABLE plocklista_rad(
   id INT,
   produkt_kod INT,
   plocklista INT,
   antal_att_plocka INT,
   plats_pa_lager VARCHAR(80),

   PRIMARY KEY (id),
   FOREIGN KEY (produkt_kod) REFERENCES produkt(produkt_kod),
   FOREIGN KEY (plats_pa_lager) REFERENCES lager(lager_hylla),
   FOREIGN KEY (plocklista) REFERENCES plocklista(plocklista_nummer)
);
```

```
CREATE TABLE faktura(
    faktura_nummer INT AUTO_INCREMENT,
    kund_id INT,
    order_nummer INT,
    totala_summan DECIMAL(6, 2),
    skapades DATETIME DEFAULT CURRENT_TIMESTAMP,
    skickades DATETIME DEFAULT NULL,

    PRIMARY KEY (faktura_nummer),
    FOREIGN KEY (order_nummer) REFERENCES ordrar(order_nummer),
    FOREIGN KEY (kund_id) REFERENCES kunder(id)
);

CREATE TABLE faktura_rad(
    id INT,
    faktura_nummmer INT,
    produkt_id INT,
    pris_per_styck DECIMAL(5, 2),
    antal INT,
    pris_per_rad DECIMAL(6, 2),

    PRIMARY KEY (id),
    FOREIGN KEY (faktura_nummmer) REFERENCES faktura(faktura_nummer),
    FOREIGN KEY (produkt_id) REFERENCES produkt(produkt_kod)
);
```