

Databasmodellering

BuckStar online shopping

2024-02-21
Av
Sara Habanakeh
& Mohamed Bnshi



TABLE OF CONTENT

Introduction	2
Konceptuell Modelling	3
Text	3
Entiteter	3
Relationer	4
ER-diagram	4
Logisk Modelling	5
ER-diagram	5
Fysisk Modelling	6
DDL SQL	7
API	7

Introduktion

En databasmodell för en e-handel (eshop) enligt kokboken som följer en väldefinierad struktur för att hantera och organisera olika aspekter av verksamheten. Modellen omfattar konceptuell, logisk och fysisk modellering, samt innehåller entiteter, relationer och SQL Data Definition Language (DDL) för att skapa de nödvändiga tabellerna.

Konceptuell Modellering

Text:

Databasen behöver hantera ett kundregister (kunder med kontakt detaljer), 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.

Entiteter:

- Kund(kund_id, namn, address)
- Produkt(produktkod_id, produktkod, produkt_namn, beskrivning, pris)
- Kategori(kategori_id, kategori_namn, produktkod)
- Lager(lager_id, hylla, produktkod, produkt_antal)
- Order(order_nr, kund_namn, kund_address, produktkod, antal_beställda)
- Plocklista(plocklista_id, order_nr, produktkod, antal_beställda, hylla)
- Faktura(faktura_nr, kund_namn, order_nr, datum, pris, total_pris)
- Logg(log_id, datum, faktura_id, handelse)

Relationer:

[1:N] varje produkt finns i en eller flera produktkategorier.

[1:N] ett lager där man ser hur många av varje produkt som finns.

[1:1] kunden beställer en produkt.

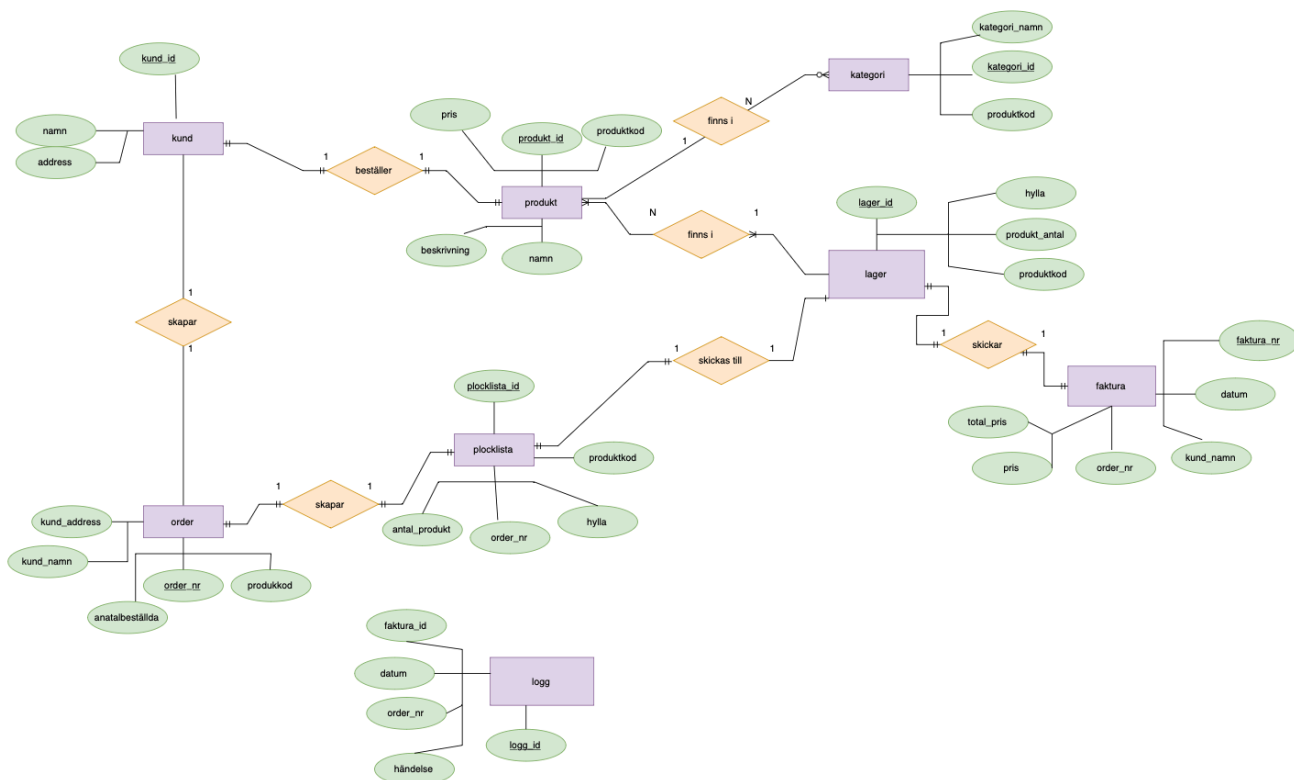
[1:1] Utifrån ordern skapas en plocklista

[1:1] plocklista som kan skickas till lagret.

[1:1] När kunden beställer en produkt så skapas en order

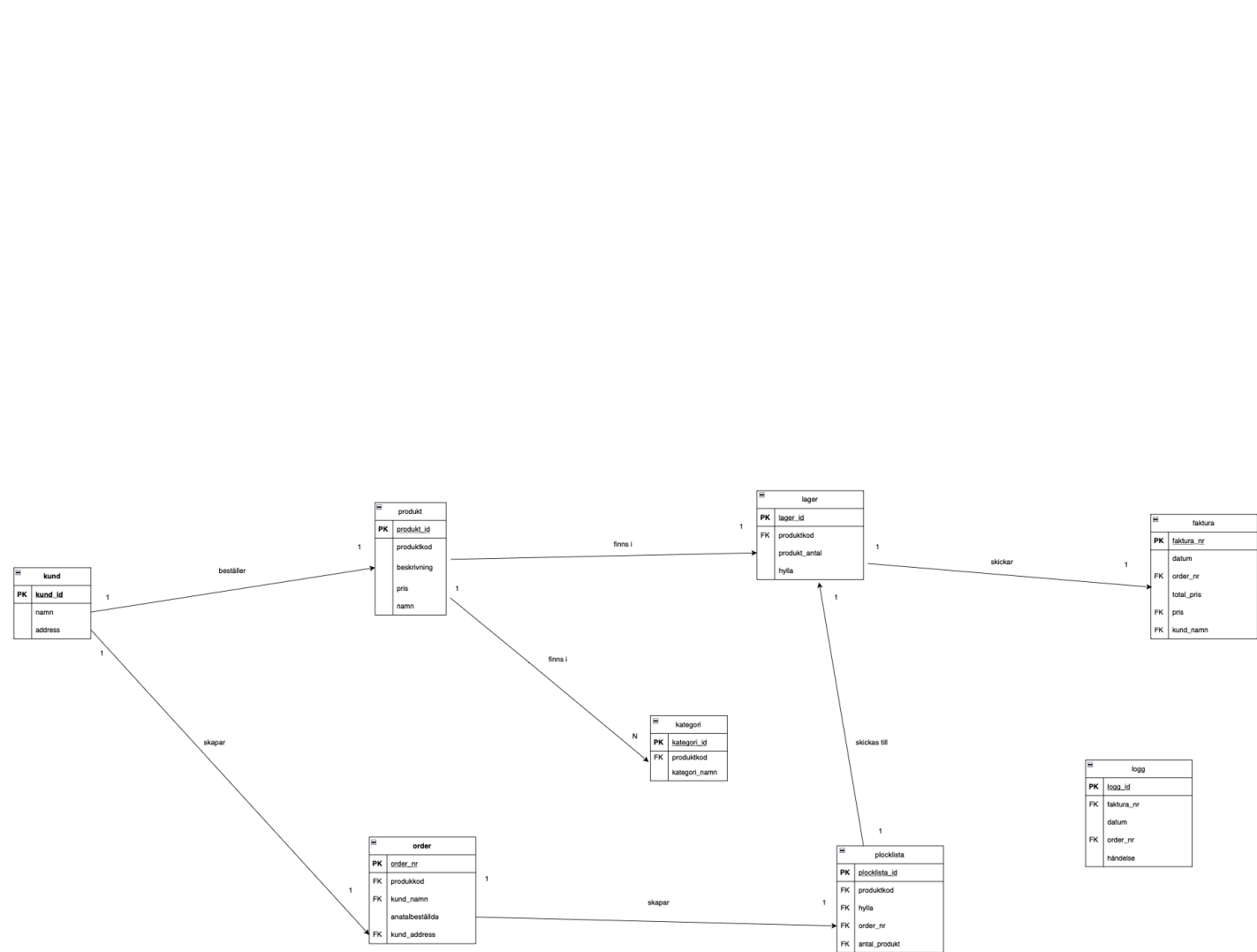
[1:1] När leveransen är packad så bifogas en faktura

ER-diagram:



Logisk Modellering

Er-diagram:



Fysisk Modelling

SQL DDL för tabellerna:

```
CREATE TABLE kund
(
  kund_id INT NOT NULL,
  namn CHAR(100),
  address VARCHAR (255),

  PRIMARY KEY (kund_id)
);
```

```
CREATE TABLE produkt
(
  produkt_id INT PRIMARY KEY NOT NULL,
  namn CHAR(100),
  produktkod CHAR(100),
  beskrivning VARCHAR (255),
  pris INT,

  PRIMARY KEY (produkt_id)
);
```

```
CREATE TABLE kategori
(
  kategori_id INT PRIMARY KEY NOT NULL,
  kategori_namn CHAR(100),
  produktkod CHAR(100),

  PRIMARY KEY (kategori_id),
  FOREIGN KEY(produktkod) REFERENCES produkt(produktkod)
);
```

```
CREATE TABLE lager
(
  lager_id INT PRIMARY KEY NOT NULL,
  produktkod CHAR(100),
  produktantal INT,
  hylla INT,

  PRIMARY KEY (lager_id),
  FOREIGN KEY(produktkod) REFERENCES produkt(produktkod)
);
```

```
CREATE TABLE order
(
  order_nr INT PRIMARY KEY NOT NULL,
  produktkod CHAR(100),
  kund_namn CHAR(100),
  kund_address VARCHAR (255),
  anatalbeställda INT,

  PRIMARY KEY (order_nr),
  FOREIGN KEY(produktkod) REFERENCES produkt(produktkod),
  FOREIGN KEY(kund_namn) REFERENCES kund(namn),
  FOREIGN KEY(kund_address) REFERENCES kund(address)
);
```

```
CREATE TABLE plocklista
(
  plocklista_id INT PRIMARY KEY NOT NULL,
  produktkod CHAR(100),
  antal_produkt INT,
  order_nr INT,
  hylla INT,

  PRIMARY KEY (plocklista_id),
  FOREIGN KEY(produktkod) REFERENCES produkt(produktkod),
  FOREIGN KEY(order_nr) REFERENCES order(order_nr),
  FOREIGN KEY(antal_produkt) REFERENCES order(anatalbeställda),
  FOREIGN KEY(hylla) REFERENCES lager(hylla)
);
```

```
CREATE TABLE faktura
(
  faktura_nr INT PRIMARY KEY NOT NULL,
  order_nr INT,
  datum INT,
  kund_namn CHAR(100),
  pris INT,
  total_pris INT,

  PRIMARY KEY (faktura_nr),
  FOREIGN KEY(pris) REFERENCES produkt(pris),
  FOREIGN KEY(order_nr) REFERENCES order(order_nr),
  FOREIGN KEY(kund_namn) REFERENCES kund(namn)
);
```



```
CREATE TABLE logg
(
  Logg_id INT PRIMARY KEY NOT NULL,
  faktura_nr INT,
  order_nr INT,
  datum INT,
  händelse VARCHAR (255)

  PRIMARY KEY (logg_id),
  FOREIGN KEY(faktura_nr) REFERENCES faktura(faktura_nr),
  FOREIGN KEY(order_nr) REFERENCES order(order_nr)
);
```

API:

- Hämta kundinformation.
- Uppdatera kundinformation.
- Hämta produktinformation.
- Uppdatera produktinformation.
- Behandla kund beställningar.
- Hantera lagerinformation.
- Generera fakturor och plocklistor.
- Sök och filtrera loggposter.

