BuckStar Online Shopping

Skapad utav Frida Persson frpe21 & Erik Nästesjö erna21 i BTHs Databas kurs

Innehållsförteckning

Konceptuell modellering

Steg	Modelleringsfas	Vad?
1	Konceptuell	Beskriv databasen i ett textstycke.
2	=	Skriv ned alla entiteter.
3	=	Skriv ned alla relationer och visa i matris.
4	=	Rita enkelt ER-diagram med entiteter och relationer.
5	=	Komplettera ER-diagram med kardinalitet.
6	=	Komplettera ER-diagram med alla attribut samt kandidatnycklar.

Textbeskrivning

Databasen hanterar ett kundregister (kunder med kontakt detaljer) samt ett produkt register (produkter med produktid, namn, kort beskrivning och pris) där varje produkt finns i en eller flera produktkategorier.

Databasen visar även lager information. Så som 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 och attribut

Kund

- Kundnamn
- Adress
- Kreditkort
- Mail
- Kund ID (Kandidatnyckel)

Produkt

- Produktkod (Kandidatnyckel)
- Produktnamn
- Produktbeskrivning
- Produktpris
- Typ

Produktkategori

- Typ
- Varumärke

Plocklista

- Kundnamn
- Kundmail
- Kundaddress
- Produktkod
- Produktnamn
- Produktbeskrivning
- Produkthylla
- Orderdatum
- Ordernummer

- Antal produkter kvar

Faktura

- Kundnamn
- Kundadress
- Kundmail
- Produktkod
- Produktnamn
- Produkbeskrivning
- Produktpris
- Orderdatum
- Ordernummer
- Antal per produkt
- Total antal produkter
- Totalpris
- Fakturanummer

Logg

- Kundnamn
- Kundadress
- Produktkod
- Produktnamn
- Produktbeskrivning
- Produktpris
- Produkthylla
- Orderdatum
- Ordernummer
- Order antal produkter
- Antal produkter i lager
- Beställningsdatum

- Fraktdatum
- Orderstatus

Relationer

- En kund kan göra en eller flera ordrar
- En order har en eller flera produkter
- En order skapar en faktura, en plocklista och en loggaktivitet
- En produkt har en eller flera produktkategorier
- En plocklista (med produkter) skickas till kund
- En plocklista skapar en loggaktivitet
- En faktura skickas till kund

Matris

Entiteter	**Kund**	**Order**	**Produkt**	**Produkt-kategori**	**Plocklista**	**Faktura**	**Logg**
Kund		skapar					
Order			har		skapar	skapar	skapar
Produkt				har			
Plocklista	skickas till						skapar
Faktura	skickas till						
Logg							

Kardinalitet

Kund - Order
- One to Many

Kund - Faktura
- One to One

Produkt - Order
- One to Many

Produkt - ProduktKategori
- One to Many

Produkt - Plocklista
- One to Many

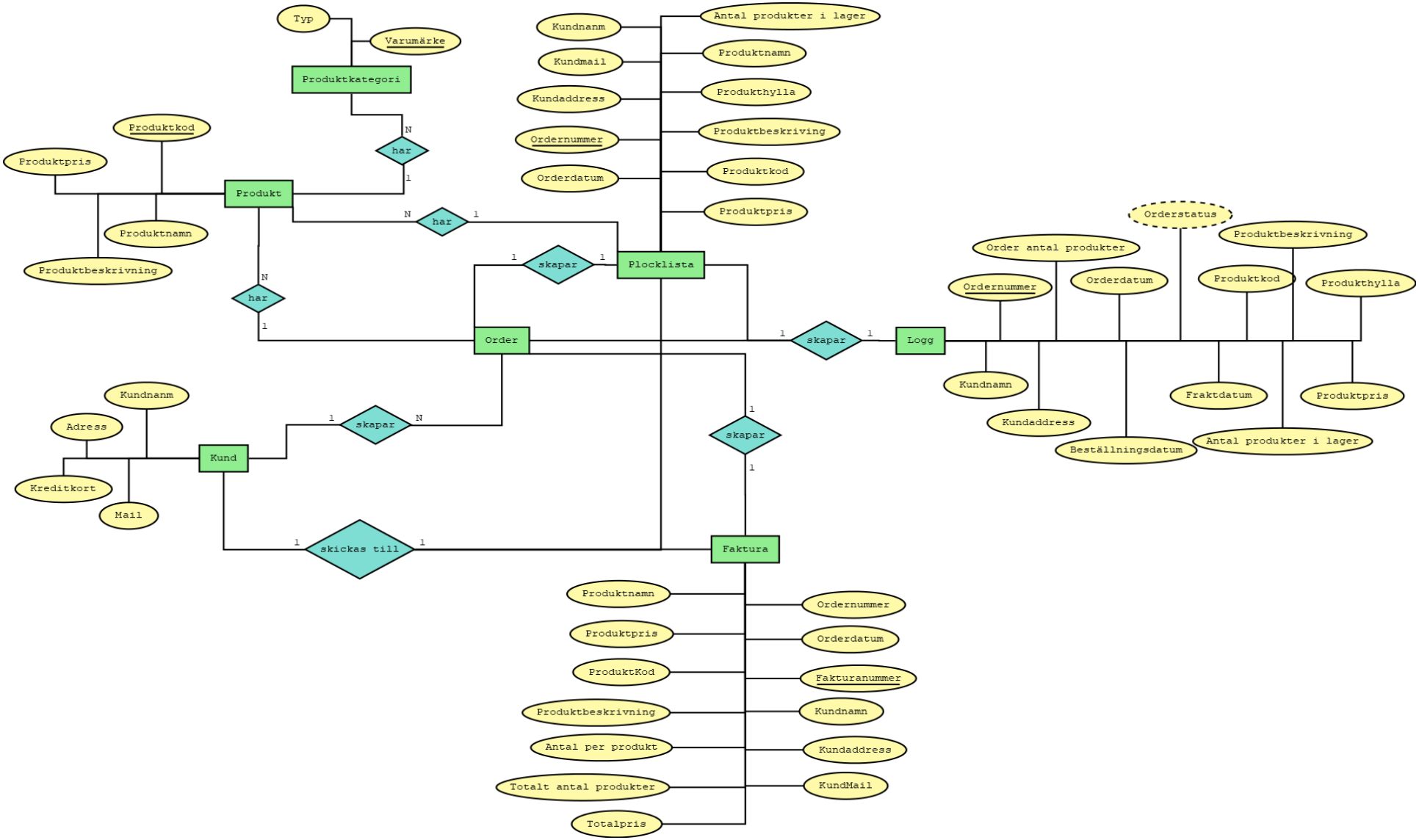
Order - Plocklista
- One to One

Order - Logg
- One to One

Order - Faktura
- One to One

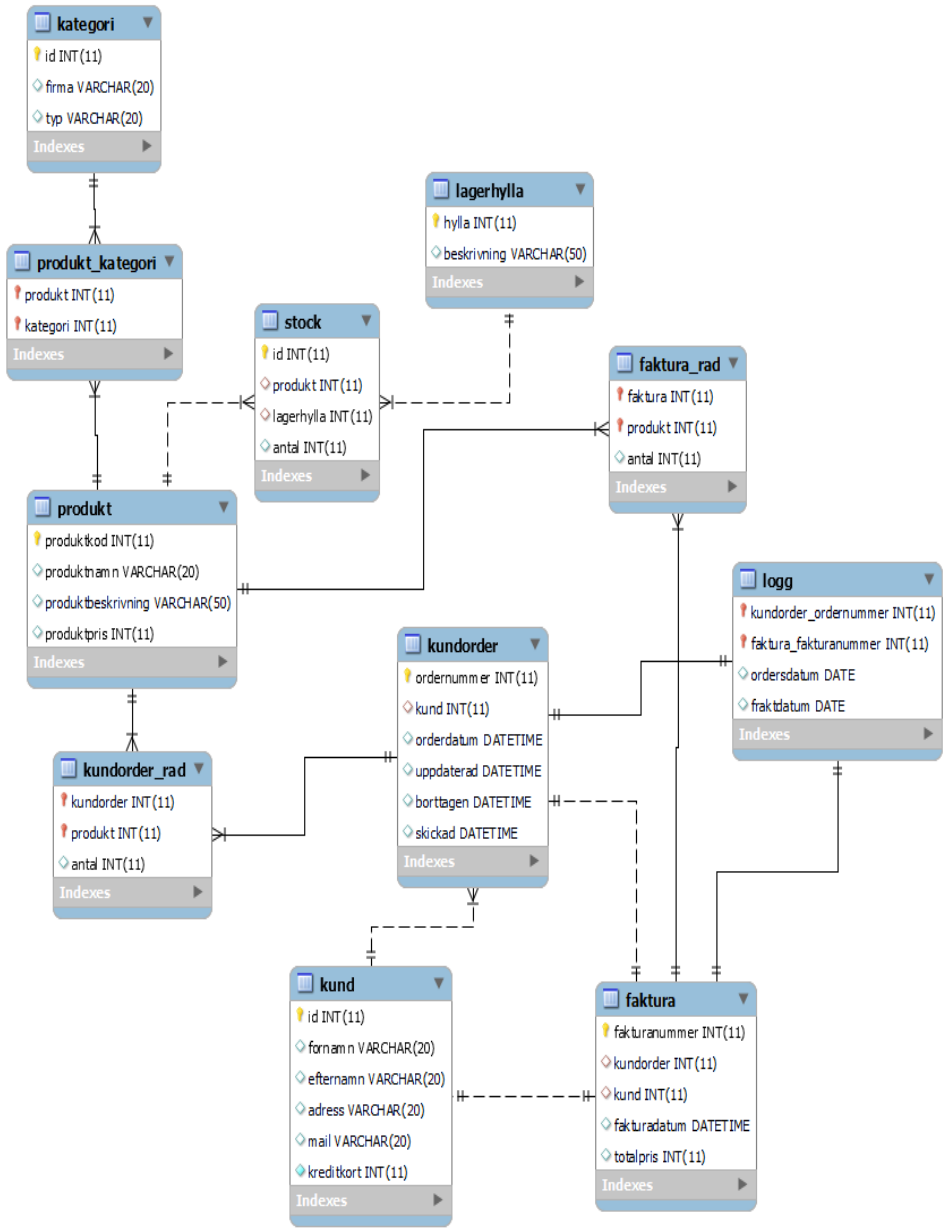
Order - Logg
- One to One

ER-diagram


Typ
Varumärke
Produktkategori
N
har
1
Produktkod
Produktpris
Produkt
Produktnamn
Produktbeskrivning
N
har
1
N
har
1
Kundnamn
Kundmail
Kundaddress
Ordernummer
Orderdatum
Antal produkter i lager
Produktnamn
Produkthylla
Produktbeskriving
Produktkod
Produktpris
1
skapar
1
Plocklista
Orderstatus
Order antal produkter
Ordernummer
Orderdatum
Produktbeskrivning
Produktkod
Produkthylla
1
skapar
1
Order
1
skapar
1
Logg
Kundnamn
Kundaddress
Beställningsdatum
Fraktdatum
Antal produkter i lager
Produktpris
Kundnamn
Adress
Kreditkort
Mail
Kund
1
skapar
N
1
skapar
1
1
skickas till
1
Faktura
Produktnamn
Produktpris
ProduktKod
Produktbeskrivning
Antal per produkt
Totalt antal produkter
Totalpris
Ordernummer
Orderdatum
Fakturanummer
Kundnamn
Kundaddress
KundMail

Logisk modellering

7	Logisk	Modifiering utav ER-diagrammet enligt relationsmodellen.
8	=	Utökning utav ER-diagram med primära/främmande nycklar samt kompletterande attribut.


kategori
id INT(11)
firma VARCHAR(20)
typ VARCHAR(20)
Indexes
produkt_kategori
produkt INT(11)
kategori INT(11)
Indexes
lagerhylla
hylla INT(11)
beskrivning VARCHAR(50)
Indexes
stock
id INT(11)
produkt INT(11)
lagerhylla INT(11)
antal INT(11)
Indexes
faktura_rad
faktura INT(11)
produkt INT(11)
antal INT(11)
Indexes
produkt
produktkod INT(11)
produktnamn VARCHAR(20)
produktbeskrivning VARCHAR(50)
produktpris INT(11)
Indexes
logg
kundorder_ordernummer INT(11)
faktura_fakturanummer INT(11)
ordersdatum DATE
fraktdatum DATE
Indexes
kundorder
ordernummer INT(11)
kund INT(11)
orderdatum DATETIME
uppdaterad DATETIME
borttagen DATETIME
skickad DATETIME
Indexes
kundorder_rad
kundorder INT(11)
produkt INT(11)
antal INT(11)
Indexes
kund
id INT(11)
fornamn VARCHAR(20)
efternamn VARCHAR(20)
adress VARCHAR(20)
mail VARCHAR(20)
kreditkort INT(11)
Indexes
faktura
fakturanummer INT(11)
kundorder INT(11)
kund INT(11)
fakturadatum DATETIME
totalpris INT(11)
Indexes

Fysisk modell

SQL DDL

Ett git repo skapades för att dela kod mellan gruppdeltagarna. SQL-koden skrevs manuellt i editorn och den logiska modellen genererades i MySQL Workbench utifrån SQL-koden.

```
-- UTAN FK sist, MED FK först
DROP TABLE IF EXISTS logg;
DROP TABLE IF EXISTS faktura_rad;
DROP TABLE IF EXISTS faktura;
DROP TABLE IF EXISTS produkt_kategori;
DROP TABLE IF EXISTS kundorder_produkt;
DROP TABLE IF EXISTS kundorder;

DROP TABLE IF EXISTS kund;
DROP TABLE IF EXISTS produkt;
DROP TABLE IF EXISTS kategori;



-----------------------------------------
-- PRODUKT
-- produkt, kategori, produkt_kategori
```

```sql
CREATE TABLE produkt
(
    produktkod INT AUTO_INCREMENT,
    produktnamn VARCHAR(20),
    produktbeskrivning VARCHAR(50),
    produktpris INT,

    PRIMARY KEY (produktkod)
);

CREATE TABLE kategori
(
    id INT AUTO_INCREMENT,
    firma VARCHAR(20),
    typ VARCHAR(20),

    PRIMARY KEY (id)
);

CREATE TABLE produkt_kategori
(
    produkt INT,
    kategori INT,

    PRIMARY KEY(produkt, kategori),
    FOREIGN KEY(produkt) REFERENCES produkt(produktkod),
    FOREIGN KEY(kategori) REFERENCES kategori(id)
```

```
);

-- ----------------------------------------------------------------
-- KUND
-- kund, kundorder, kundorder_rad
--

CREATE TABLE kund
(
    id INT,
    fornamn VARCHAR(20),
    efternamn VARCHAR(20),
    adress VARCHAR(20),
    mail VARCHAR(20),
    kreditkort INT NOT NULL,

    PRIMARY KEY (id)
);

CREATE TABLE kundorder
(
    ordernummer INT AUTO_INCREMENT,
    kund INT,
    orderdatum DATETIME DEFAULT CURRENT_TIMESTAMP,
    uppdaterad DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
    borttagen DATETIME DEFAULT NULL,
    skickad DATETIME DEFAULT NULL,
```

```
    PRIMARY KEY (ordernummer),
    FOREIGN KEY (kund) REFERENCES kund(id)
);

CREATE TABLE kundorder_rad
(
    kundorder INT,
    produkt INT,
    antal INT,
    PRIMARY KEY (kundorder, produkt),
    FOREIGN KEY (kundorder) REFERENCES kundorder(ordernummer),
    FOREIGN KEY (produkt) REFERENCES produkt(produktkod)
);

-----------------------------------------------------------------
-- FAKTURA
-- faktura - faktura_rad

CREATE TABLE faktura
(
    fakturanummer INT,
    kundorder INT,
    kund INT,
    fakturadatum DATETIME DEFAULT CURRENT_TIMESTAMP,
    totalpris INT,
```

```
    PRIMARY KEY(fakturanummer),
    FOREIGN KEY(kundorder) REFERENCES kundorder(ordernummer),
    FOREIGN KEY(kund) REFERENCES kund(id)
);

CREATE TABLE faktura_rad
(
    faktura INT,
    produkt INT,
    antal INT,
    PRIMARY KEY (faktura, produkt),
    FOREIGN KEY(produkt) REFERENCES produkt(produktkod),
    FOREIGN KEY(faktura) REFERENCES faktura(fakturanummer)
);

-------------------------------------------------------------------------
-- LAGER
-- lagerhylla - logg - stock

CREATE TABLE lagerhylla
(
    hylla INT,
    beskrivning VARCHAR(50),

    PRIMARY KEY(hylla)
);
```

```sql
CREATE TABLE stock
(
    id INT AUTO_INCREMENT,
    produkt INT,
    lagerhylla INT,
    antal INT,

    PRIMARY KEY (id),
    FOREIGN KEY (produkt) REFERENCES produkt(produktkod),
    FOREIGN KEY (lagerhylla) REFERENCES lagerhylla(hylla)
);


CREATE TABLE logg
(
    kundorder_ordernummer INT,
    faktura_fakturanummer INT,
    ordersdatum DATE,
    fraktdatum DATE,

    PRIMARY KEY(kundorder_ordernummer, faktura_fakturanummer),
    FOREIGN KEY(kundorder_ordernummer) REFERENCES kundorder(ordernummer),
    FOREIGN KEY(faktura_fakturanummer) REFERENCES faktura(fakturanummer)
);

----------------------
-- VIEWS
```

```sql
-- plocklista

DROP VIEW IF EXISTS plocklista;

CREATE VIEW plocklista
AS
SELECT
    ko.ordernummer,
    ko.kund,
    kor.kundorder,
    kor.produkt,
    kor.antal
FROM kundorder as ko
    INNER JOIN kundorder_rad AS kor
        on ko.ordernummer = kor.kundorder
;

SHOW TABLES;

DESCRIBE plocklista;
```

Lista funktioner som databasen skall stödja (API)

1. Lägg till order
2. Uppdatera order
3. Radera order
4. Visa order
5. Visa kund
6. Visa faktura
7. Visa stock
8. Visa logg
9. Sök information ang. produkter
10. Uppdatera produktantal vid köp