Kaktusfarmen AB

# Databasmodell

Kmom03 Databas

Rasmus Steen
2018-02-17

# Innehåll

Kaktusfarmen AB ..... 2
Konceptuell modell ..... 2
Databasen ..... 2
Tabeller ..... 2
Relationer ..... 2
Diagram ..... 3
Kardinalitet ..... 4
Attribut ..... 5
Logisk Modell ..... 6
Tabeller ..... 6
Fysisk Modell ..... 7
SQL ..... **Fel! Bokmärket är inte definierat.**
Stödjer ..... 9

# Kaktusfarmen AB

Kaktusfarmen är ett företag som säljer mestadels kaktusar som dem själva odlat i växthus här i Sverige. Dem säljer även en del palmer, krukor och planteringsjord. Kaktusfarmen vill nu ta sig an den digitala marknaden och starta en eshop där det går att beställa deras produkter.

# Konceptuell modell

## Databasen

Databasen behöver kunna hantera kunder med tillhörande information för leverans av produkten. En kund måste också kunna lägga en order och få en faktura. Produkter har en kategori och finns planterade i en planteringsyta i ett växthus. Kunden vill kunna se en sammanställning av vad som finns i alla växthus så dem vet hur mycket dem har på lager av varje produkt. Några av planteringsytorna planteras det inget i utan där förvaras jord eller krukor. Produkter har kategorier t.ex kaktus, kruka eller annan växt. Personalen måste också veta vad som ska med i ett paket och var den produkten finns.

## Tabeller

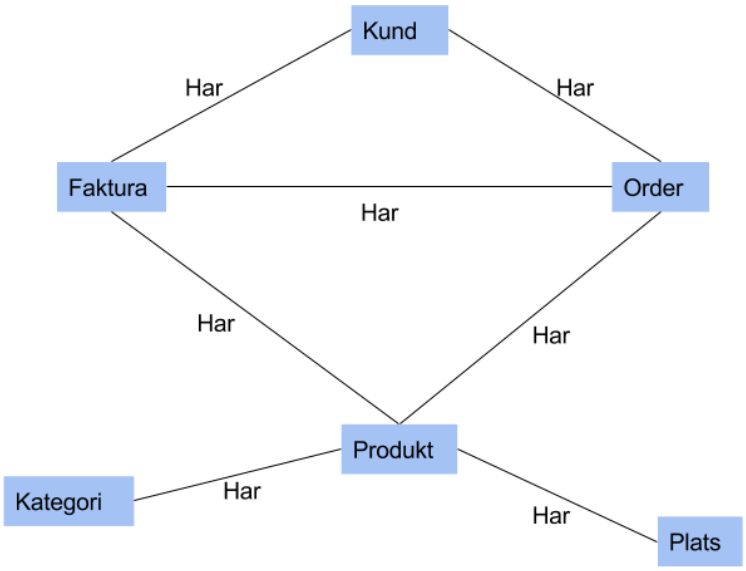
- Kunder
- Order
- Faktura
- Produkter
- Kategorier
- Plats (produktens plats bland växthus och planteringsytor)
- Paket (plocklista för personal)

## Relationer

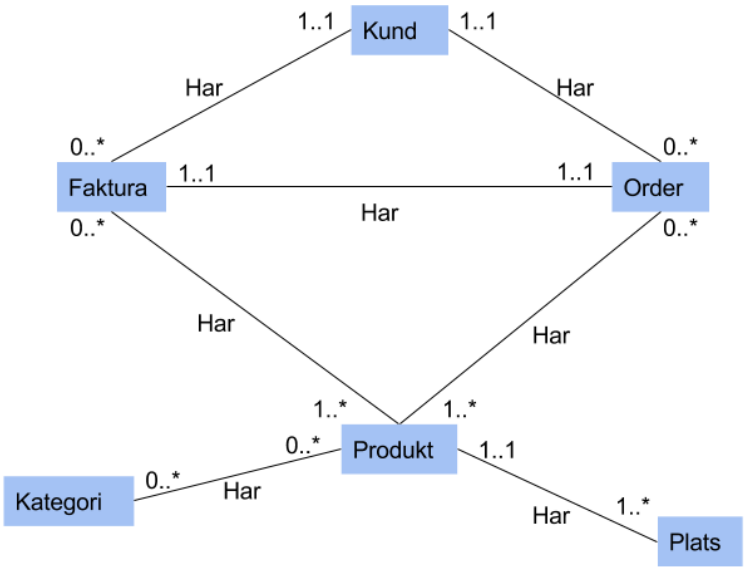
- Alla kunder har ett antal orders, fakturor och paket
- En order har ett antal produkter
- En faktura har en order och ett antal produkter
- En produkt har flera kategorier och kan finnas på flera platser
- En kategori kan ha flera produkter
- En plats kan innehålla en produkt
- Ett paket har en order och flera produkter och platser

| Entiteter | **Kund** | **Order** | **Faktura** | **Produkt** | **Kategori** | **Plats** |
|---|---|---|---|---|---|---|
| **Kund** | | Flera | Flera | | | |
| **Order** | | | | Flera | | |
| **Faktura** | | En | | Flera | | |
| **Produkt** | | | | | Flera | Flera |
| **Kategori** | | | | Flera | | |
| **Plats** | | | | En | | |

## Diagram

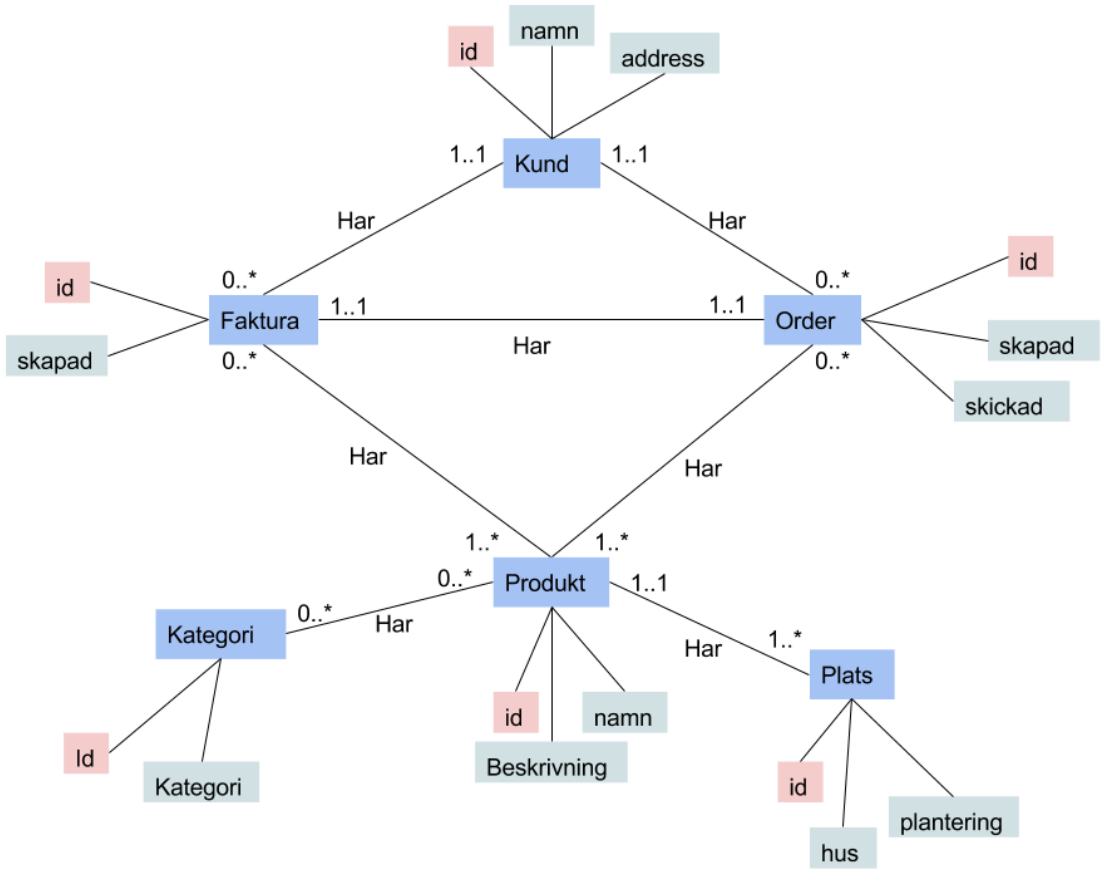

Kund
Har
Har
Faktura
Order
Har
Har
Har
Produkt
Kategori
Har
Har
Plats

## Kardinalitet




| **Relation** | **Typ** |
|---|---|
| Kund - Faktura | 1:N |
| Kund - Order | 1:N |
| Faktura - Order | 1:1 |
| Faktura - Produkt | N:M |
| Order - Produkt | N:M |
| Produkt - Kategori | N:M |
| Produkt - Plats | 1:N |

## Attribut



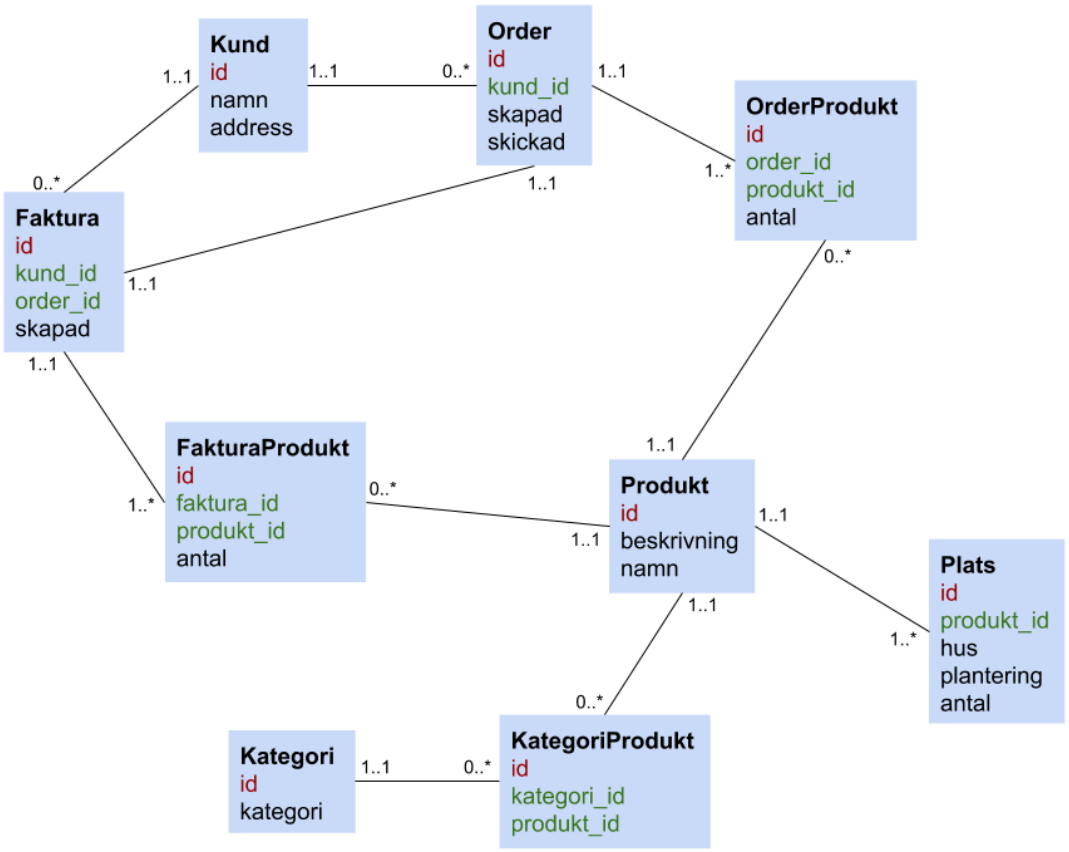

- Kund (id, namn, adress)
- Faktura (id, skapad)
- Order (id, skapad, skickad)
- Produkt (id, namn, beskrivning)
- Kategori (id, kategori)
- Plats (id, hus, plantering)

## Logisk Modell

Har tagit bort många till många förhållanden:

- Faktura – Produkt
- Order – Produkt
- Produkt – Kategori

I bilden nedan gäller: Fetstil = Tabellnamn, Röd text = Primär nyckel, Grön text = Foreign key




## Tabeller

- Kund (id, namn, address)
- Produkt (id, beskrivning, namn)
- Kategori (id, kategori)
- Order (id, #kund_id, skapad, skickad)
- Faktura (id, #kund_id, #order_id, skapad)
- Plats (id, #produkt_id, hus, plantering, antal)
- OrderProdukt (id, #order_id, #produkt_id, antal)
- FakturaProdukt (id, #faktura_id, #produkt_id, antal)
- KategoriProdukt (id, #kategori_id, #produkt_id)

# Fysisk Modell

## DDL

Sql:en skrevs för hand, det var inte så mycket.

```
-- Non-depending tables
CREATE TABLE IF NOT EXISTS kund
(
      id INT NOT NULL,
    namn VARCHAR (64) NOT NULL,
    address VARCHAR (96) NOT NULL,

    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS produkt
(
      id INT NOT NULL,
    namn VARCHAR (64) NOT NULL,
    beskrivning VARCHAR (128) NOT NULL,

    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS kategori
(
      id INT NOT NULL,
    kategori VARCHAR (64) NOT NULL UNIQUE,

    PRIMARY KEY (id)
);

-- Depending on non-depending
CREATE TABLE IF NOT EXISTS bestallning
(
      id INT NOT NULL,
    kund_id INT NOT NULL,
    skapad DATE NOT NULL,
    skickad DATE,

    PRIMARY KEY (id),
    FOREIGN KEY (kund_id) REFERENCES kund(id)
);

CREATE TABLE IF NOT EXISTS plats
(
      id INT AUTO_INCREMENT,
    produkt_id INT NOT NULL,
    hus VARCHAR (16) NOT NULL,
    plantering INT NOT NULL,
    antal INT NOT NULL,

    PRIMARY KEY (id),
    FOREIGN KEY (produkt_id) REFERENCES produkt(id)
);
```

```
CREATE TABLE IF NOT EXISTS kategoriprodukt
(
      id INT AUTO_INCREMENT,
    produkt_id INT NOT NULL,
    kategori_id INT NOT NULL,

    PRIMARY KEY (id),
    FOREIGN KEY (produkt_id) REFERENCES produkt(id),
    FOREIGN KEY (kategori_id) REFERENCES kategori(id)
);

-- Depending on depending
CREATE TABLE IF NOT EXISTS faktura
(
      id INT NOT NULL,
    kund_id INT NOT NULL,
    bestallning_id INT NOT NULL,
    skapad DATE NOT NULL,

    PRIMARY KEY (id),
    FOREIGN KEY (kund_id) REFERENCES kund(id),
    FOREIGN KEY (bestallning_id) REFERENCES bestallning(id)
);

CREATE TABLE IF NOT EXISTS bestallningprodukt
(
      id INT AUTO_INCREMENT,
    produkt_id INT NOT NULL,
    bestallning_id INT NOT NULL,
    antal INT,

    PRIMARY KEY (id),
    FOREIGN KEY (produkt_id) REFERENCES produkt(id),
    FOREIGN KEY (bestallning_id) REFERENCES bestallning(id)
);

CREATE TABLE IF NOT EXISTS fakturaprodukt
(
      id INT AUTO_INCREMENT,
    produkt_id INT NOT NULL,
    faktura_id INT NOT NULL,
    antal INT,

    PRIMARY KEY (id),
    FOREIGN KEY (produkt_id) REFERENCES produkt(id),
    FOREIGN KEY (faktura_id) REFERENCES faktura(id)
);
```

## Stödjer

Lista över funktioner som databasen skall ha stöd för

- Skapa en kund
- Skapa en beställning
- Skapa en faktura
- Lägga till produkter
- Lägga till kategorier
- Visa antalet av en specifik produkt
- Visa produkter
- Visa fakturor för en kund
- Visa beställningar för en kund
- Söka efter produkt
- Söka efter produkt via kategori
- Vista platsen för en produkt i lagret
- Reducera antalet av en produkt i lagret