

# ER-modell till

CdOff online shopping



I denna modellering kommer jag att beskriva databasen, specificerar alla delar av databasen ihop med deras relation till varandra och till slut presentera en visuell ritning som förklarar databasens struktur.

Denna modellering är gjort av Paul Moreland

# Innehåll

3.      Konceptuell modell
4.      ER-diagram av databasen
5.      Logisk modell
6.      Fysisk modell
7.      Appendix

# Konceptuell modell

## Beskrivning av databasen

Vi ska utveckla en e-shop där fokus ligger på försäljning av CD/DVD och äldre vinyl-skivor samt relaterande produkter. E-shopen kommer att snyggt presentera dessa produkter tillsammans med en kort beskrivning och pris samt är sorterat i olika kategorier.

E-shopen kommer att innehålla ett lagersystem där man kan se hur många av varje produkt finns kvar i lager och vilken hylla produkten finns på.

Varje kund som besöker e-shopen kommer ha möjlighet att köpa dessa produkter där de behöver registrerar sig med namn och kontaktdetaljer. När kunden beställer en produkt så skapas en order där plocklistan skickas till lagret för hantering av lagerpersonalen och sedan förbereds för leverans. I packningsprocessen så bifogas en faktura som innehåller en prisspecifisering för varje produkt i ordern och det totala priset.

Det kommer att finnas ett log system som lagrar alla viktiga händelser i systemet, som tex. beställningar och order ändringar.

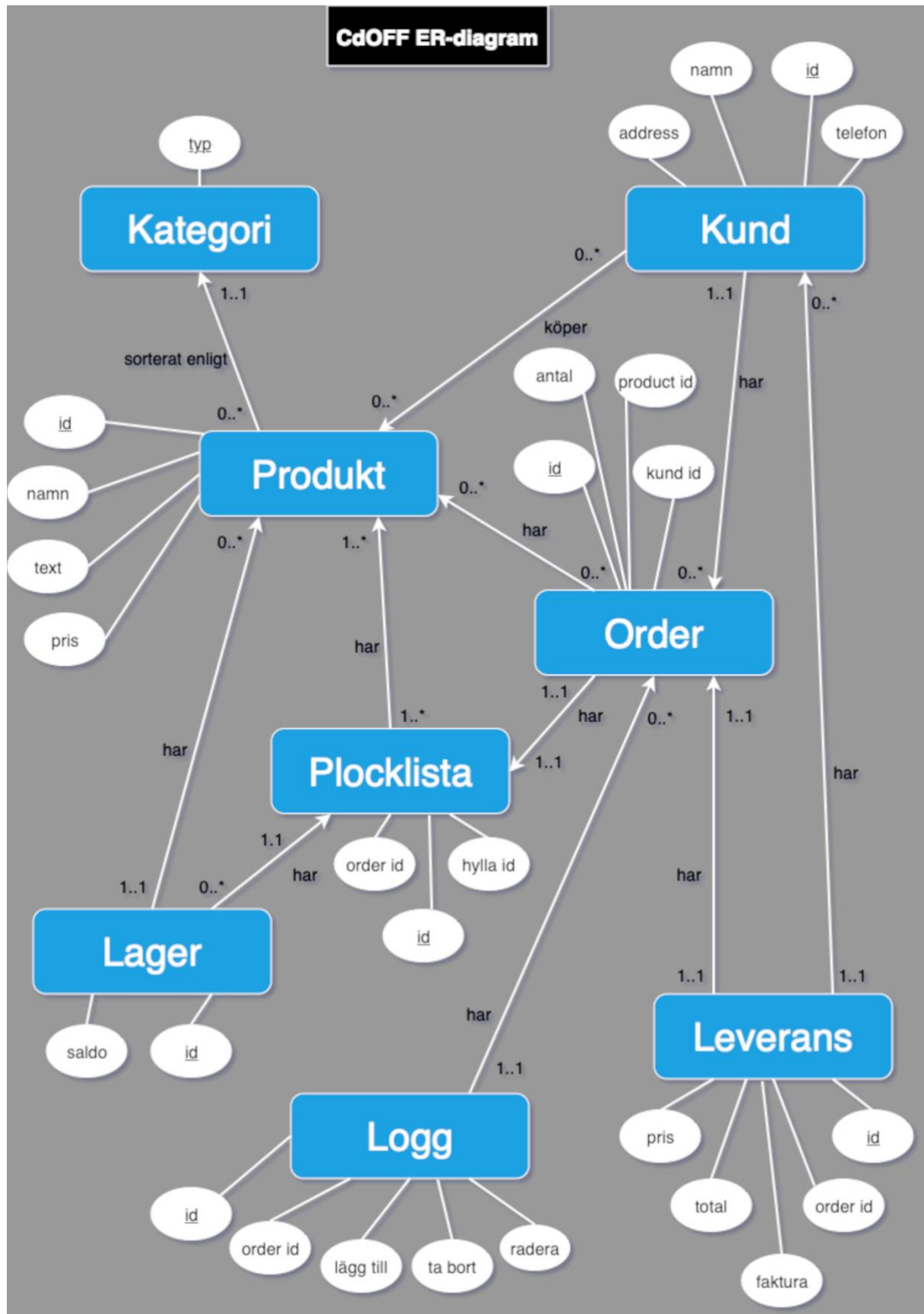
## Alla entiteter

- Produkt
- Kund
- Kategori
- Lager
- Order
- Plocklista
- Leverans
- logg

## Alla Relationer, läs vänster till höger

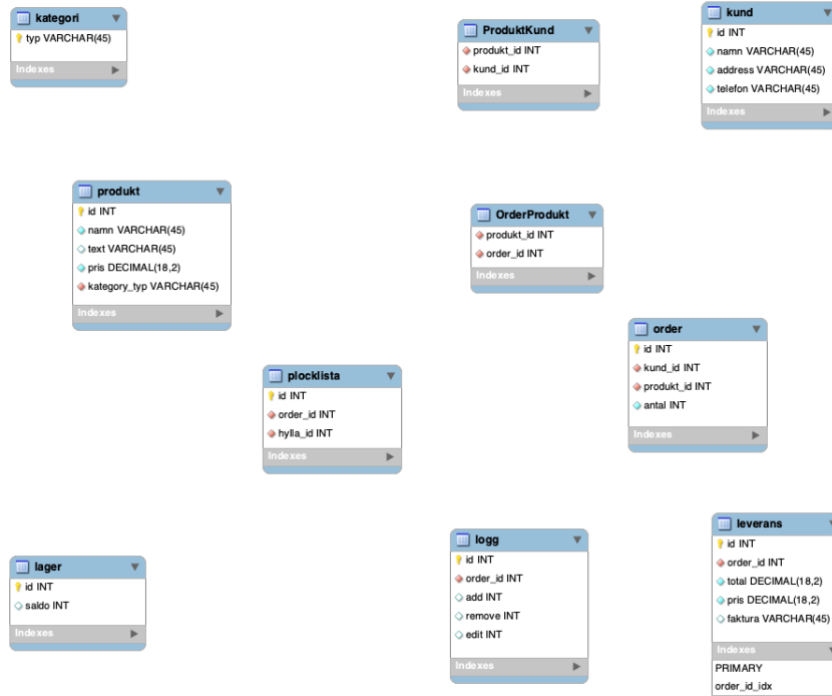
Entiteter	Produkt	Kategori	Kund	Order	Plocklista	Lager	Leverans	Logg
Produkt		sorterat enligt	är köpt av	läggs i	läggs i	finns i		
Kategori	sorterar							
Kund	köper			har			få	
Order	har		Skapats av		har		Skickas med	
Plocklista	har			Skapat i		Läsas av		
Lager	har				har			
Leverans			har	har				
Logg				har				

# ER-diagram



# Logisk modell

## ER-diagram enligt Relationsmodellen



### Text beskrivning

**produkt** (id, namn, text, pris, kategori\_typ)

**kund** (id, namn, address, telefon)

**kategori** (typ)

**ProduktKund** (produkt\_id, kund\_id)

**OrderProdukt** (produkt\_id, order\_id)

**order** (id, kund\_id, produkt\_id, antal)

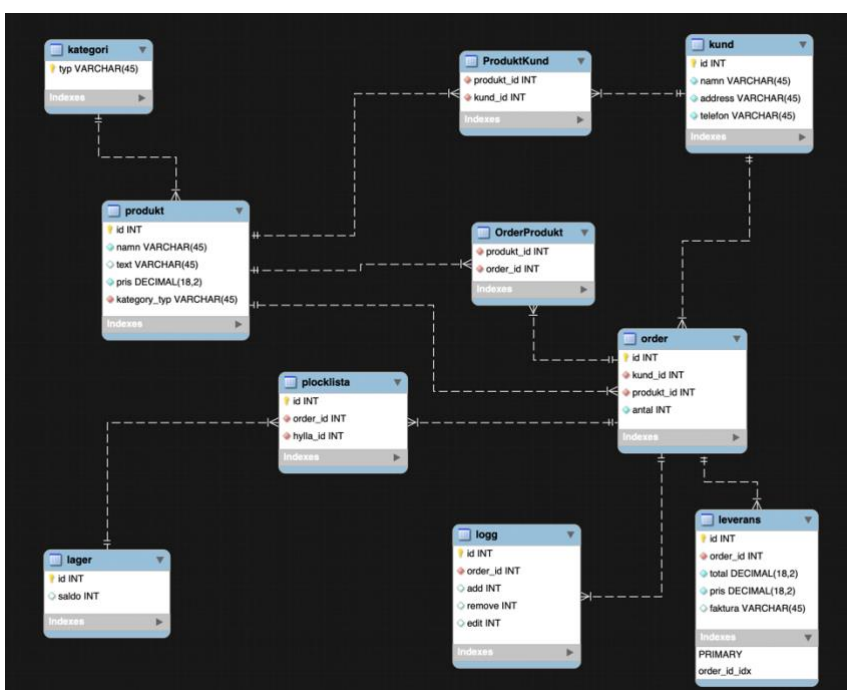
**plocklista** (id, order\_id, hylla\_id)

**lager** (id, saldo)

**leverans** (id, order\_id, total, pris, faktura)

**logg** (id, order\_id, add, remove, edit)

## Utökat ER-diagram med primära och främmande nycklar samt kompletterande attribut



### Text beskrivning

#### Primära (\*) & Främmande (#) nycklar

produkt (\*id, #kategori\_typ)

kund (\*id)

kategori (\*typ)

produktKund (#produkt\_id, #kund\_id)

OrderProdukt (#produkt\_id, #order\_id)

order (\*id, #kund\_id, #produkt\_id)

plocklista (\*id, #order\_id, #hylla\_id)

lager (\*id)

leverans (\*id, #order\_id)

logg (\*id, #order\_id)

# Fysisk modell

## Databas

MySQL

## SQL DDL

Jag använde MySQLWorkbench för att generera SQL koden till databasen, [se här](#).

## API – Funktioner som databasen skall stödja

1. Lägg till produkt i lager med hyllplats nr
2. Uppdatera produkt
3. Radera produkt
4. Visa alla produkter & information
5. Sök produkter
6. Lägg till kund
7. Uppdatera kund
8. Radera kund
9. Visa alla kunder & detaljer
10. Sök kund
11. Kund skapar order med produkter
12. Plocklista från ordern skapats ihop med produkternas hyllplats nr
13. Lager saldo justeras
14. Faktura med pris skapats
15. Färdig plockat order med faktura läggs i leverans till kund
16. Viktiga händelser läggs i logg:
  - 16.1. Kund skapar order
  - 16.2. Plocklista skapats
  - 16.3. Lager saldo ändrats
  - 16.4. Plocklista raderas
  - 16.5. Leverans skapats
  - 16.6. Faktura skapats

# Appendix

```
-- -----  
-- Table `eshop`.`kategori`  
-- -----
```

```
DROP TABLE IF EXISTS `eshop`.`kategori` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`kategori` (  
  `typ` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`typ`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `eshop`.`produkt`  
-- -----
```

```
DROP TABLE IF EXISTS `eshop`.`produkt` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`produkt` (  
  `id` INT NOT NULL,  
  `namn` VARCHAR(45) NOT NULL,  
  `text` VARCHAR(45) NULL,  
  `pris` DECIMAL(18,2) NOT NULL,  
  `kategori_typ` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `kategori_typ_idx` (`kategori_typ` ASC) VISIBLE,  
  CONSTRAINT `kategori_typ`  
    FOREIGN KEY (`kategori_typ`)  
      REFERENCES `eshop`.`kategori` (`typ`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `eshop`.`kund`  
-- -----
```

```
DROP TABLE IF EXISTS `eshop`.`kund` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`kund` (  
  `id` INT NOT NULL,  
  `namn` VARCHAR(45) NOT NULL,  
  `address` VARCHAR(45) NOT NULL,  
  `telefon` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-- Table `eshop`.`order`
```

```
DROP TABLE IF EXISTS `eshop`.`order` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`order` (  
  `id` INT NOT NULL,  
  `kund_id` INT NOT NULL,  
  `produkt_id` INT NOT NULL,  
  `antal` INT NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `kund_id_idx` (`kund_id` ASC) VISIBLE,  
  INDEX `produkt_id_idx` (`produkt_id` ASC) VISIBLE,  
  CONSTRAINT `kund_id_order`  
    FOREIGN KEY (`kund_id`)  
      REFERENCES `eshop`.`kund` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `produkt_id_order`  
    FOREIGN KEY (`produkt_id`)  
      REFERENCES `eshop`.`produkt` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `eshop`.`lager`
```

```
DROP TABLE IF EXISTS `eshop`.`lager` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`lager` (  
  `id` INT NOT NULL,  
  `saldo` INT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```



```
-- Table `eshop`.`plocklista`
```

```
DROP TABLE IF EXISTS `eshop`.`plocklista` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`plocklista` (  
  `id` INT NOT NULL,  
  `order_id` INT NOT NULL,  
  `hylla_id` INT NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `order_id_idx` (`order_id` ASC) VISIBLE,  
  INDEX `hylla_id_idx` (`hylla_id` ASC) VISIBLE,  
  CONSTRAINT `order_id_pl`  
    FOREIGN KEY (`order_id`)  
      REFERENCES `eshop`.`order` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `hylla_id_pl`  
    FOREIGN KEY (`hylla_id`)  
      REFERENCES `eshop`.`lager` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `eshop`.`leverans`
```

```
DROP TABLE IF EXISTS `eshop`.`leverans` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`leverans` (  
  `id` INT NOT NULL,  
  `order_id` INT NOT NULL,  
  `total` DECIMAL(18,2) NOT NULL,  
  `pris` DECIMAL(18,2) NOT NULL,  
  `faktura` VARCHAR(45) NULL,  
  PRIMARY KEY (`id`),  
  INDEX `order_id_idx` (`order_id` ASC) VISIBLE,  
  CONSTRAINT `order_id_lv`  
    FOREIGN KEY (`order_id`)  
      REFERENCES `eshop`.`order` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `eshop`.`logg`  
-- -----
```

```
DROP TABLE IF EXISTS `eshop`.`logg` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`logg` (  
  `id` INT NOT NULL,  
  `order_id` INT NOT NULL,  
  `add` INT NULL,  
  `remove` INT NULL,  
  `edit` INT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `order_id_idx` (`order_id` ASC) VISIBLE,  
  CONSTRAINT `order_id_log`  
    FOREIGN KEY (`order_id`)  
      REFERENCES `eshop`.`order` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `eshop`.`ProduktKund`  
-- -----
```

```
DROP TABLE IF EXISTS `eshop`.`ProduktKund` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`ProduktKund` (  
  `produkt_id` INT NOT NULL,  
  `kund_id` INT NOT NULL,  
  INDEX `produkt_id_idx` (`produkt_id` ASC) VISIBLE,  
  INDEX `kund_id_idx` (`kund_id` ASC) VISIBLE,  
  CONSTRAINT `produkt_id_pk`  
    FOREIGN KEY (`produkt_id`)  
      REFERENCES `eshop`.`produkt` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `kund_id_pk`  
    FOREIGN KEY (`kund_id`)  
      REFERENCES `eshop`.`kund` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `eshop`.`OrderProdukt`
```

```
DROP TABLE IF EXISTS `eshop`.`OrderProdukt` ;
```

```
CREATE TABLE IF NOT EXISTS `eshop`.`OrderProdukt` (  
  `produkt_id` INT NOT NULL,  
  `order_id` INT NOT NULL,  
  INDEX `produkt_id_idx` (`produkt_id` ASC) VISIBLE,  
  INDEX `order_id_idx` (`order_id` ASC) VISIBLE,  
  CONSTRAINT `produkt_id_op`  
    FOREIGN KEY (`produkt_id`)  
      REFERENCES `eshop`.`produkt` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `order_id_op`  
    FOREIGN KEY (`order_id`)  
      REFERENCES `eshop`.`order` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```