

CdOff online shopping, ER-modell för en databas.

Johan Hanses, johv18.

Innehållsförteckning

Innehållsförteckning	2
Konceptuell modell	3
Databasens syfte	3
Entiteter	4
ER-diagram med attribut och kandidatnycklar	5
Logisk modell	6
Modifiera diagrammet enligt relationsmodellen	6
Utöka och komplettera ER-diagrammet	7
Fysisk modell	9
Skapa SQL DDL för tabellerna	9
Funktioner som databasen skall stödja (API)	9
sql.ddl	10

Konceptuell modell

Databasens syfte

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.

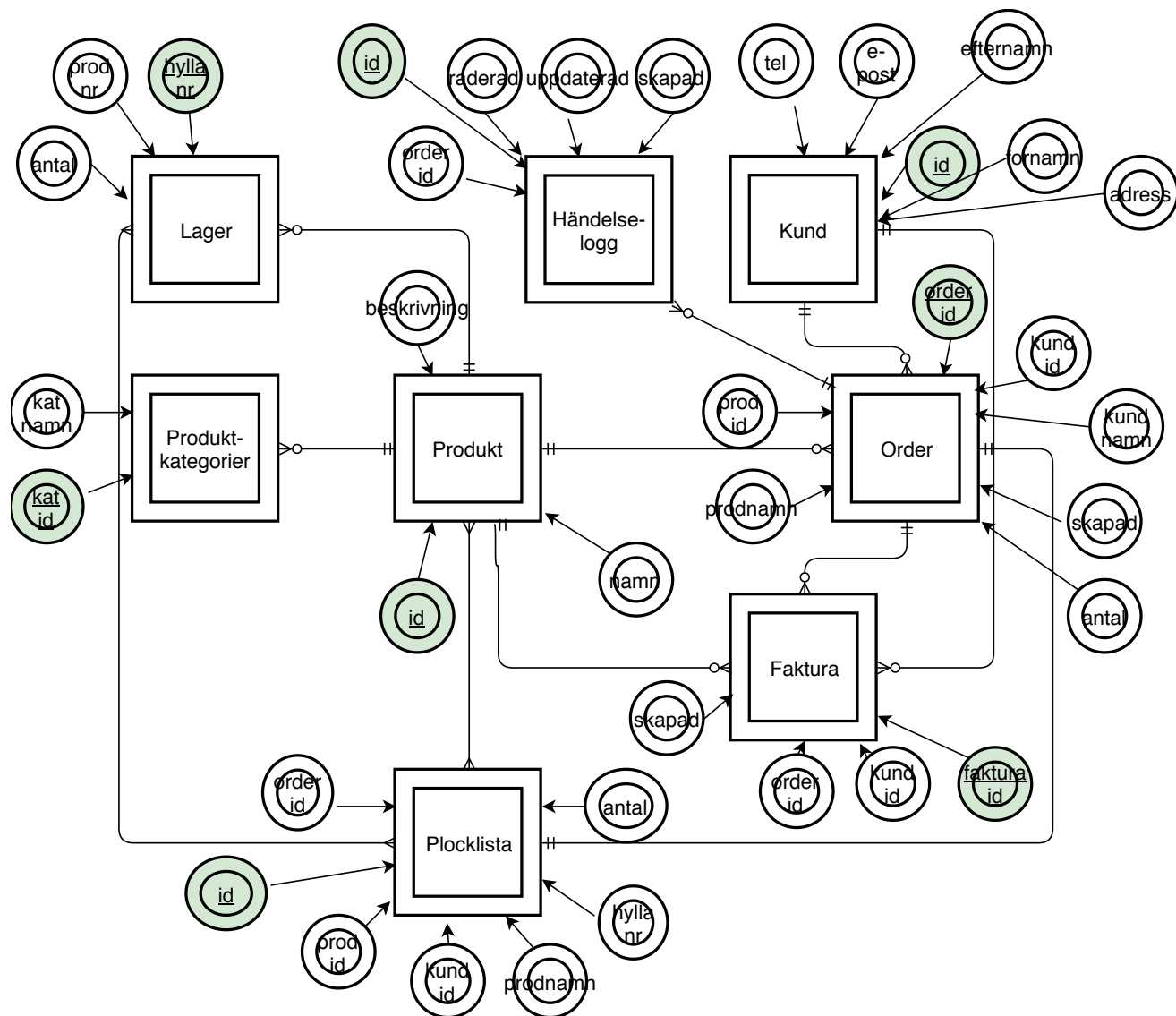
Det skall finnas en **logg** där man kan se viktiga händelser i systemet, till exempel när en order skapas, raderas, eller när det läggs till/tas bort en orderrad (en produkt) i ordern.

Entiteter

- Kund
- Produkt
- Produktkategorier
- Lager
- Order
- Plocklista
- Faktura
- Händelselogg

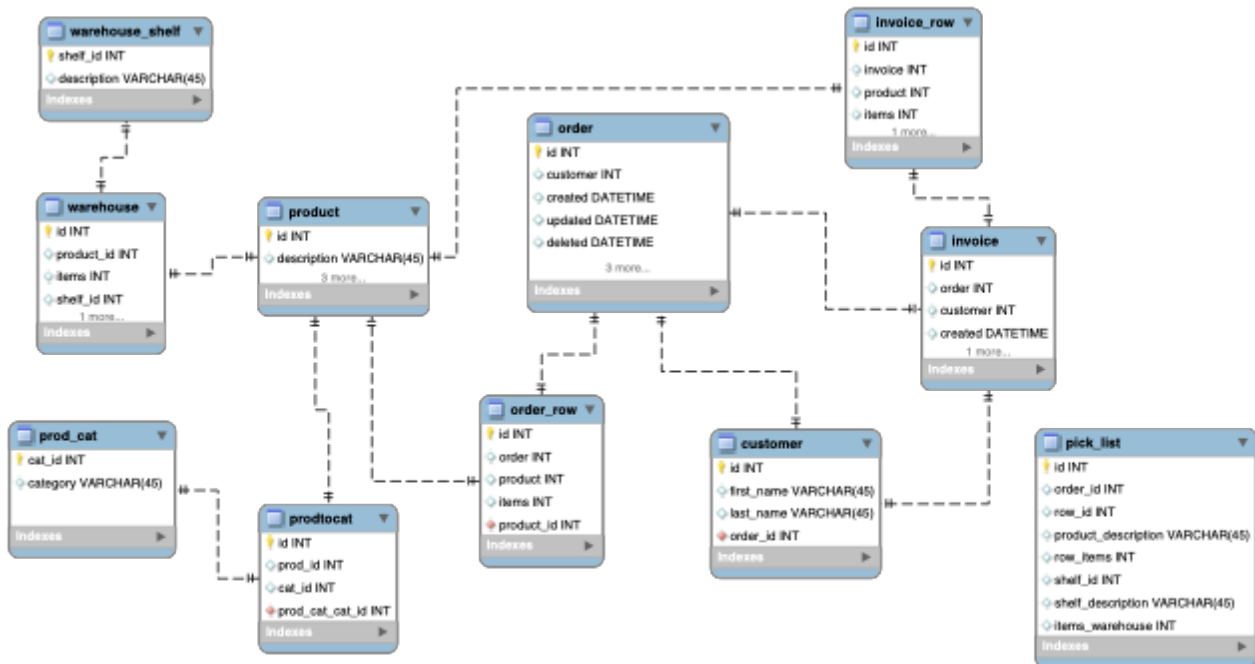
	Kund	Produkt	Produkt-kategorier	Lager	Order	Plocklista	Faktura	Händelselogg
Kund	-				har en / flera		har en / flera	
Produkt		-	har en / flera	har en / flera	har en / flera	har flera	har en / flera	
Produktkategorier		har en / flera	-					
Lager		har en / flera		-		har flera		
Order	har en / flera	har en / flera			-	har en / flera	har en / flera	har en / flera
Plocklista		har flera		har flera	har en / flera	-		
Faktura	har en / flera	har en / flera			har en / flera		-	
Händelselogg					har en / flera			-

ER-diagram med attribut och kandidatnycklar



Logisk modell

Modifiera diagrammet enligt relationsmodellen



ER-diagrammet uppdaterat enligt relationsmodellen

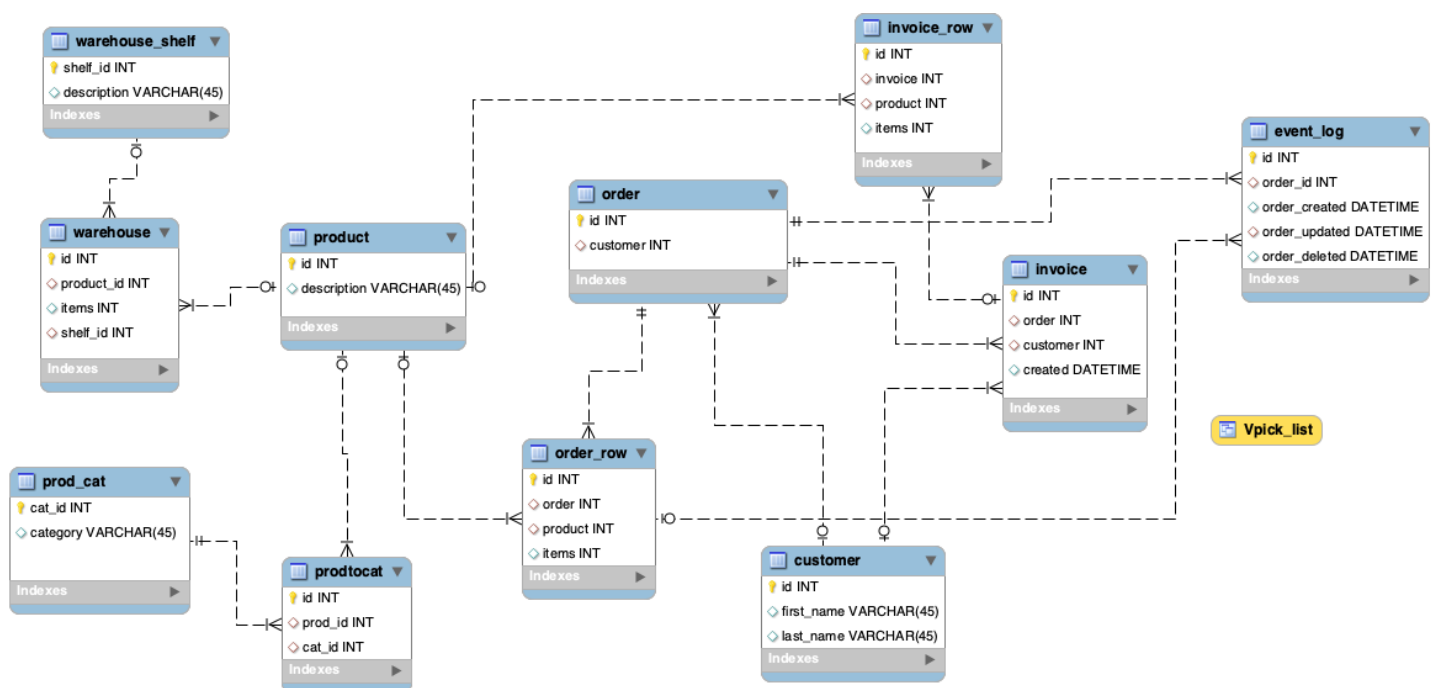
Entiteterna är nu på god väg att bli tabeller och en vy, även om allt inte är på plats ännu.

Tabellerna och kolumnnamnen är nu på engelska enligt nedan.

- warehouse (id, product_id, shelf_id, items)
- warehouse_shelf (shelf_id, description)
- order (id, customer_id, created, updated, deleted)
- order_row (id, order, produkt, items)
- customer (id, first_name, last_name)
- invoice (id, order, customer, created)
- invoice_row (id, invoice, product, items)

- picklist, kommer bli en vy med kolumner från de andra tabellerna enda unika kolumnen blir dess (id)
- event_log, ej på bild men kommer i nästa steg. även den en vy med kolumner från de andra tabellerna, enda unika kolumnen blir dess (id).
- prod_cat (cat_id, category)
- producttocat (id, prod_id, cat_id)
- product (id, description)

Utöka och komplettera ER-diagrammet



Tabellerna är nu genomarbetade och event_log är nu en egen tabell och en del kolumner har bytt plats. Tabellerna är uppdaterade med primärnyckel på de understrukna kolumnsnamnen nedan, och främmande nycklar markeras med # framför kolumnsnamnen

- warehouse (id, #product_id, #shelf_id, items)
- warehouse_shelf (shelf_id, description)
- order (id, #customer,)
- order_row (id, #order, #produkt, items)
- customer (id, first_name, last_name)
- invoice (id, #order, #customer, created)
- invoice_row (id, #invoice, #product, items)
- prod_cat (cat_id, category)
- producttocat (id, #prod_id, #cat_id)
- product (id, description)
- event_log, (id, #order_id, order_created, order_updated, order_deleted).
- Vpick_list, en vy med kolumner från de andra tabellerna.

Fysisk modell

Skapa SQL DDL för tabellerna

Jag modellerade de sista stegen av er-diagrammet i MySQLWorkbench främst för att programmet gjorde det enkelt att skapa ett bra diagram med kolumnnamn och tabellernas relation. I mappen /kmom04/er2 finns filerna setup.sql och ddl.sql. setup.sql skapar databasen eshop för användaren "user" med lösenordet "pass". Bägge filerna är skapade för hand av den enkla anledningen att det är roligast och känns mer kontrollerat. Koden i ddl.sql finns i slutet av denna rapport.

Funktioner som databasen skall stödja (API)

- Lägga en order
- Lägga till en produkt
- Ta bort en order
- Ta bort en produkt
- Uppdatera en order/produktrad

ddl.sql

```
--

-- create ddl for database eshop

--

SET NAMES 'utf8';

-- drop the tables if they exist

DROP TABLE IF EXISTS `event_log`;

DROP TABLE IF EXISTS `warehouse`;

DROP TABLE IF EXISTS `warehouse_shelf`;

DROP TABLE IF EXISTS `invoice_row`;

DROP TABLE IF EXISTS `invoice`;

DROP TABLE IF EXISTS `order_row`;

DROP TABLE IF EXISTS `order`;

DROP TABLE IF EXISTS `customer`;

DROP TABLE IF EXISTS `producttocat`;

DROP TABLE IF EXISTS `prod_cat`;

DROP TABLE IF EXISTS `product`;


-- create product related tables

CREATE TABLE `product`

(

    `id` INT NOT NULL,

    `description` VARCHAR(20),


    PRIMARY KEY (`id`)

)

ENGINE INNODB
```

CHARSET utf8

COLLATE utf8_swedish_ci;

CREATE TABLE `prod_cat`

(

 `id` INT NOT NULL,

 `category` VARCHAR(45),

 PRIMARY KEY (`id`)

)

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

CREATE TABLE `producttocat`

(

 `id` INT NOT NULL,

 `prod_id` INT,

 `cat_id` INT,

 PRIMARY KEY (`id`),

 FOREIGN KEY (`prod_id`) REFERENCES `product`(`id`),

 FOREIGN KEY (`cat_id`) REFERENCES `prod_cat`(`id`)

)

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

```

-----

-- create customer table

CREATE TABLE `customer`

(
    `id` INT NOT NULL,

    `first_name` CHAR(20),

    `last_name` CHAR(20),


    PRIMARY KEY (`id`)
)

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

-----

-- create order tables

CREATE TABLE `order`

(
    `id` INT NOT NULL,

    `customer_id` INT,


    PRIMARY KEY (`id`),

    FOREIGN KEY (`customer_id`) REFERENCES `customer`(`id`)
)

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

```

```

CREATE TABLE `order_row`
(
    `id` INT NOT NULL,
    `order` INT,
    `product` INT,
    `items` INT,

    PRIMARY KEY (`id`),
    FOREIGN KEY (`order`) REFERENCES `order`(`id`),
    FOREIGN KEY (`product`) REFERENCES `product`(`id`)
)
ENGINE INNODB
CHARSET utf8
COLLATE utf8_swedish_ci;

-----

-- create invoice tables

CREATE TABLE `invoice`
(
    `id` INT NOT NULL,
    `order` INT,
    `customer` INT,
    `created` DATETIME,

    PRIMARY KEY (`id`),
    FOREIGN KEY (`order`) REFERENCES `order`(`id`),
    FOREIGN KEY (`customer`) REFERENCES `customer`(`id`)
)

```

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

CREATE TABLE `invoice_row`

(

 `id` INT NOT NULL,

 `invoice` INT,

 `product` INT,

 `items` INT,

 PRIMARY KEY (`id`),

 FOREIGN KEY (`invoice`) REFERENCES `invoice` (`id`),

 FOREIGN KEY (`product`) REFERENCES `product` (`id`)

)

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

-- create Warehouse tables

CREATE TABLE `warehouse_shelf`

(

 `shelf_id` INT NOT NULL,

 `description` VARCHAR(45),

 PRIMARY KEY (`shelf_id`)

)

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

CREATE TABLE `warehouse`

(

 `id` INT NOT NULL,

 `product_id` INT,

 `shelf_id` INT,

 `items` INT,

 PRIMARY KEY (`id`),

 FOREIGN KEY (`product_id`) REFERENCES `product` (`id`),

 FOREIGN KEY (`shelf_id`) REFERENCES `warehouse_shelf` (`shelf_id`)

)

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

-- create event_log table

CREATE TABLE `event_log`

(

 `id` INT NOT NULL,

 `order_id` INT,

 `order_row_id` INT,

 `order_created` DATETIME,

 `order_updated` DATETIME,

```

`order_deleted` DATETIME,

PRIMARY KEY (`id`),

FOREIGN KEY (`order_id`) REFERENCES `order`(`id`),

FOREIGN KEY (`order_row_id`) REFERENCES `order_row`(`id`)

)

ENGINE INNODB

CHARSET utf8

COLLATE utf8_swedish_ci;

```

```

-----

-- create view event_log

DROP VIEW IF EXISTS `Vpick_list`;

CREATE VIEW `Vpick_list`

AS

SELECT

    `o`.`id` AS `order_number`,

    `or`.`id` AS `order_row`,

    `p`.`description` AS `description`,

    `or`.`items` AS `items_ordered`,

    `ws`.`shelf_id` AS `shelf`,

    `ws`.`description` AS `shelf_location`,

    `w`.`items` AS `items_avaliable`

FROM `order` AS `o`

    INNER JOIN `order_row` AS `or`

        ON `o`.`id` = `or`.`order`

    INNER JOIN `product` AS `p`

        ON `or`.`product` = `p`.`id`

```



```
INNER JOIN `warehouse` AS `w`  
  ON `p`.`id` = `w`.`product_id`  
INNER JOIN `warehouse_shelf` AS `ws`  
  ON `w`.`shelf_id` = `ws`.`shelf_id`  
GROUP BY `order_row`  
;
```