

DOCUMENTATION

FULLY DRESSED USE-CASE

Use-case: Process Order

Omfang og niveau: Dette omhandler en ordre der går gennem systemet

Primær aktør: Sælger

Pre betingelser: At der ikke er registret kunder

Post betingelser: Varen bliver leveret

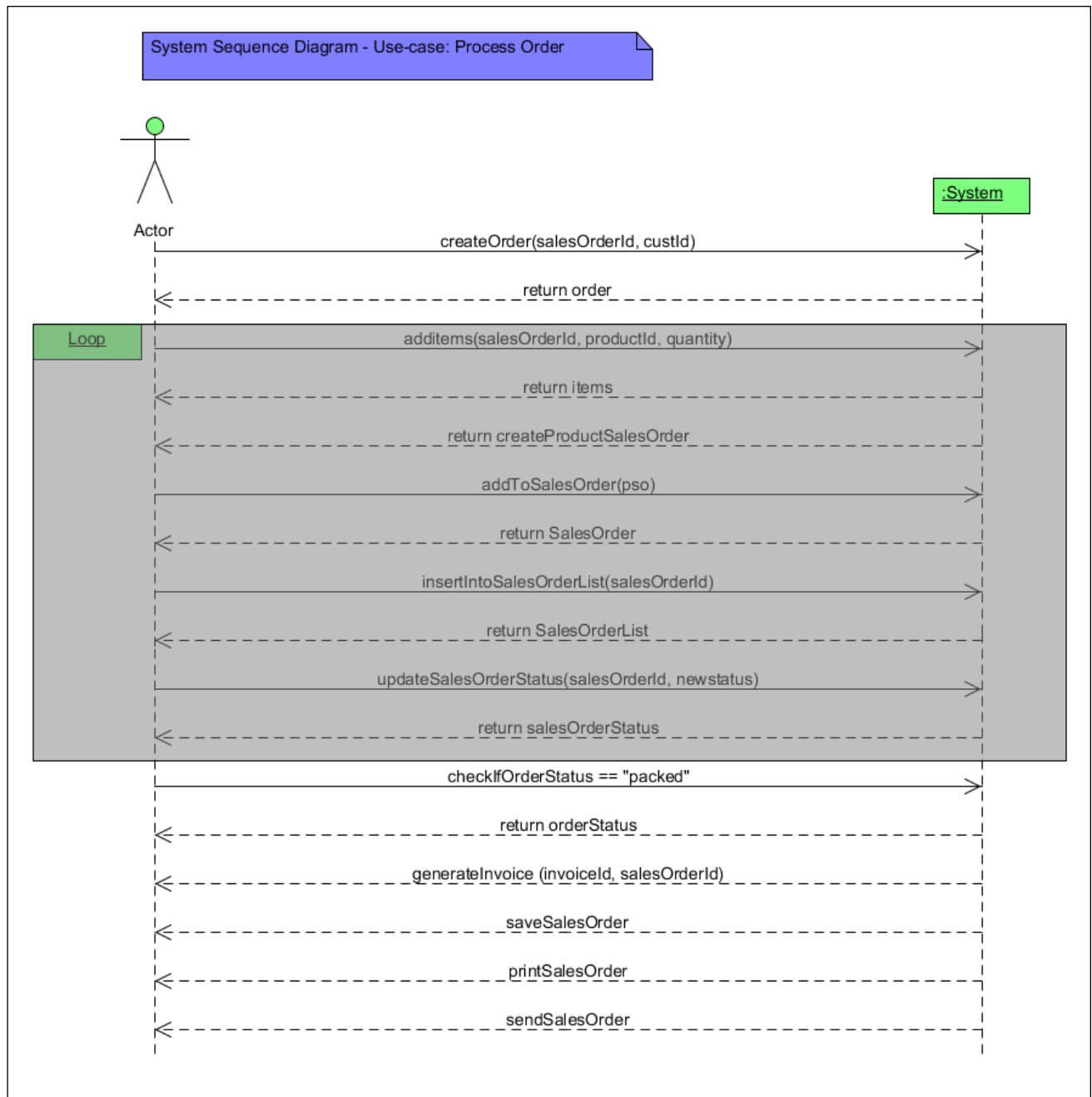
Basis succes flow:

1. Sælger modtager en ordre på mail eller telefon
2. Systemet opretter en ordre
3. Sælgeren modtager og angiver kundens oplysninger
4. Sælger tilføjer varen og antal til ordren
5. Systemet returnere at varen er tilføjet til ordren
6. Sælgeren opdaterer ordren med den nye status
7. Sælgeren modtager kundens betalingsoplysninger
8. Systemet bekræfter kundens oplysninger
9. Sælgeren afslutter ordren
10. Systemet gemmer og printer en faktura ud
11. Systemet sender en mail med fakturaen med ordreinformation til kunden

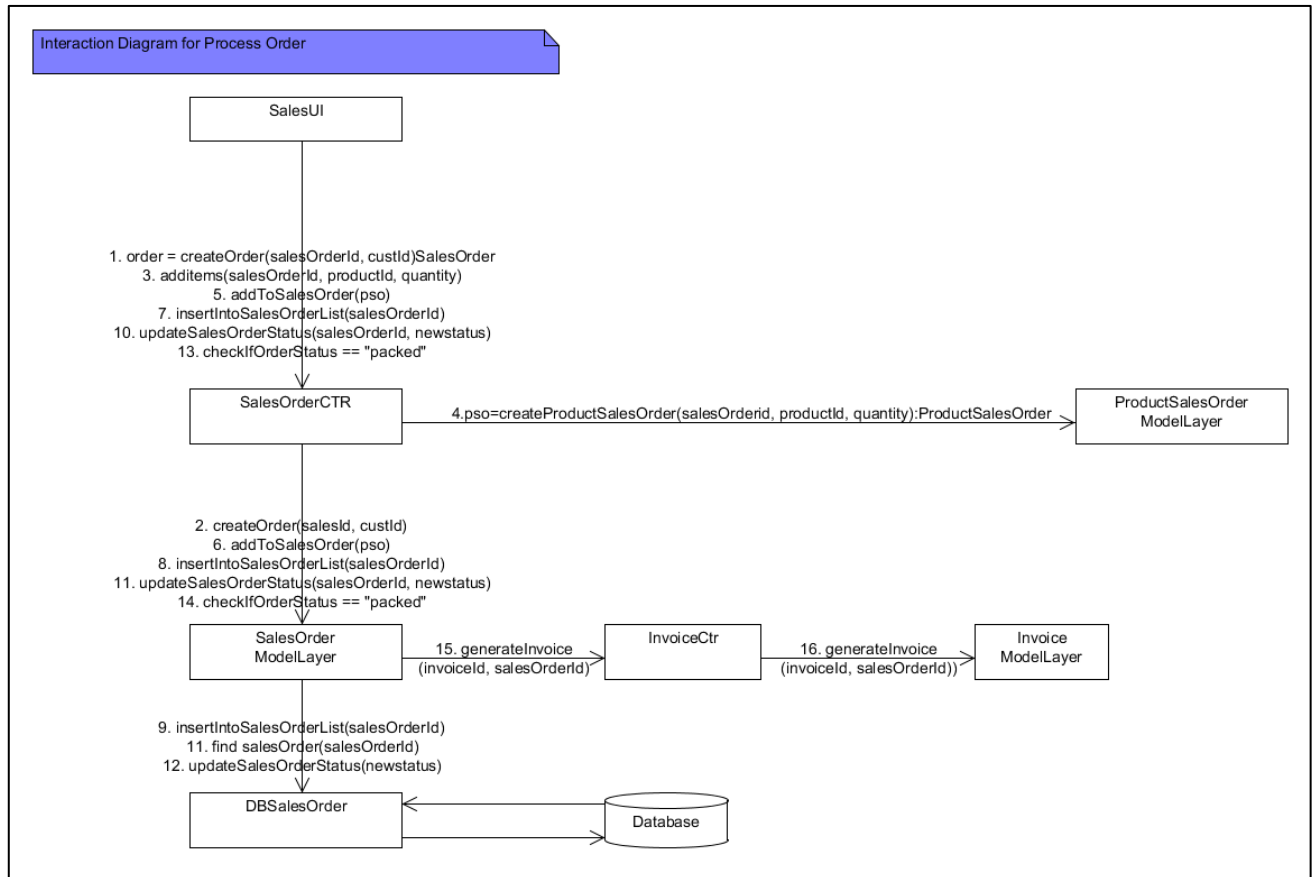
Alternativt flow:

- ❖ Systemet går ned
 - ❖ Kunden fortryder
- 4a Sælgeren indtaster et forkert varenummer
- 4b Varenummeret findes ikke i systemet
- 10a Ordren bliver ikke gemt

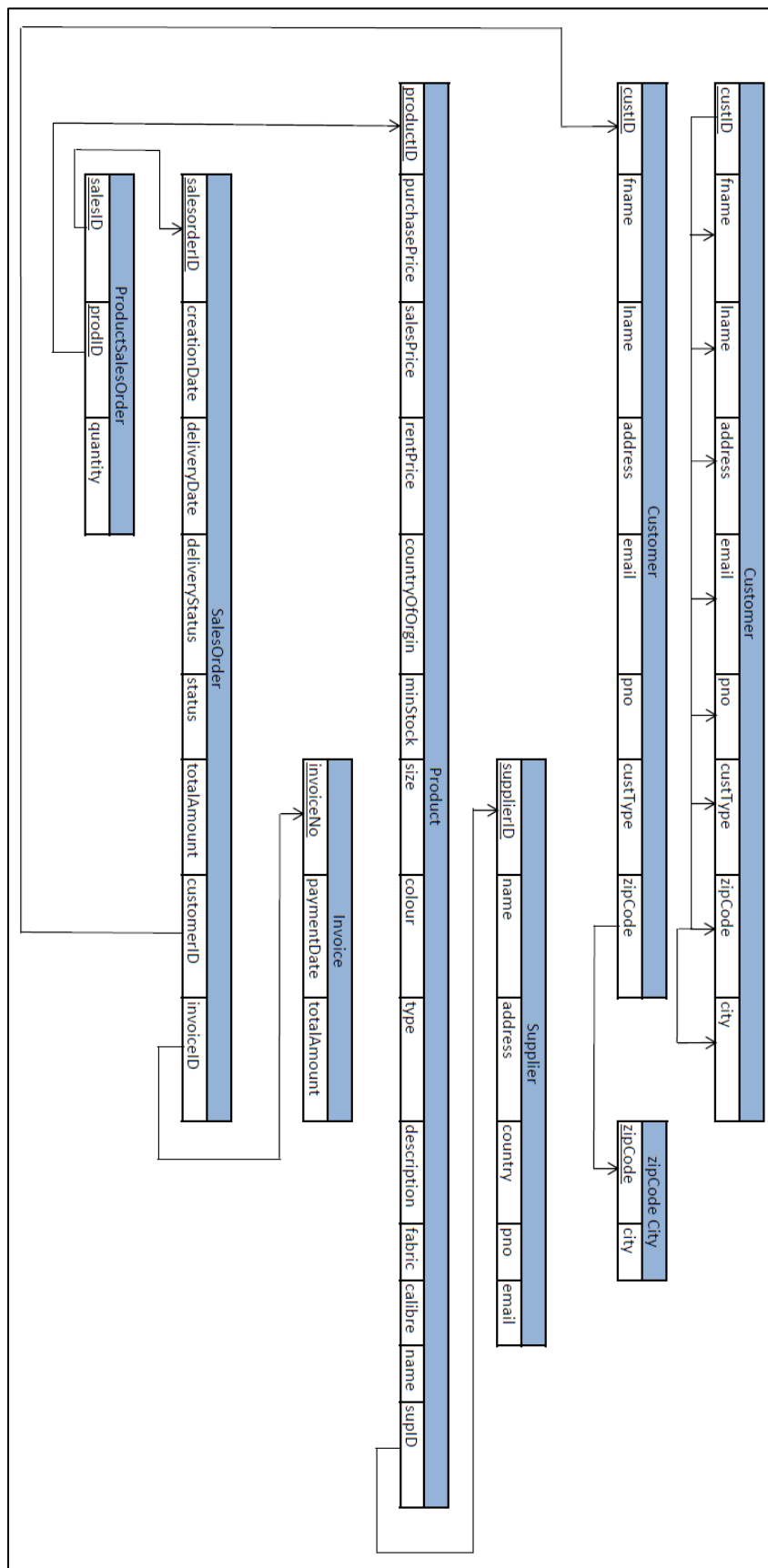
SYSTEM SEQUENCE DIAGRAM



INTERACTION DIAGRAM



RELATIONAL DATABASE SCHEMA



DATABASE SCHEMA DESCRIPTION

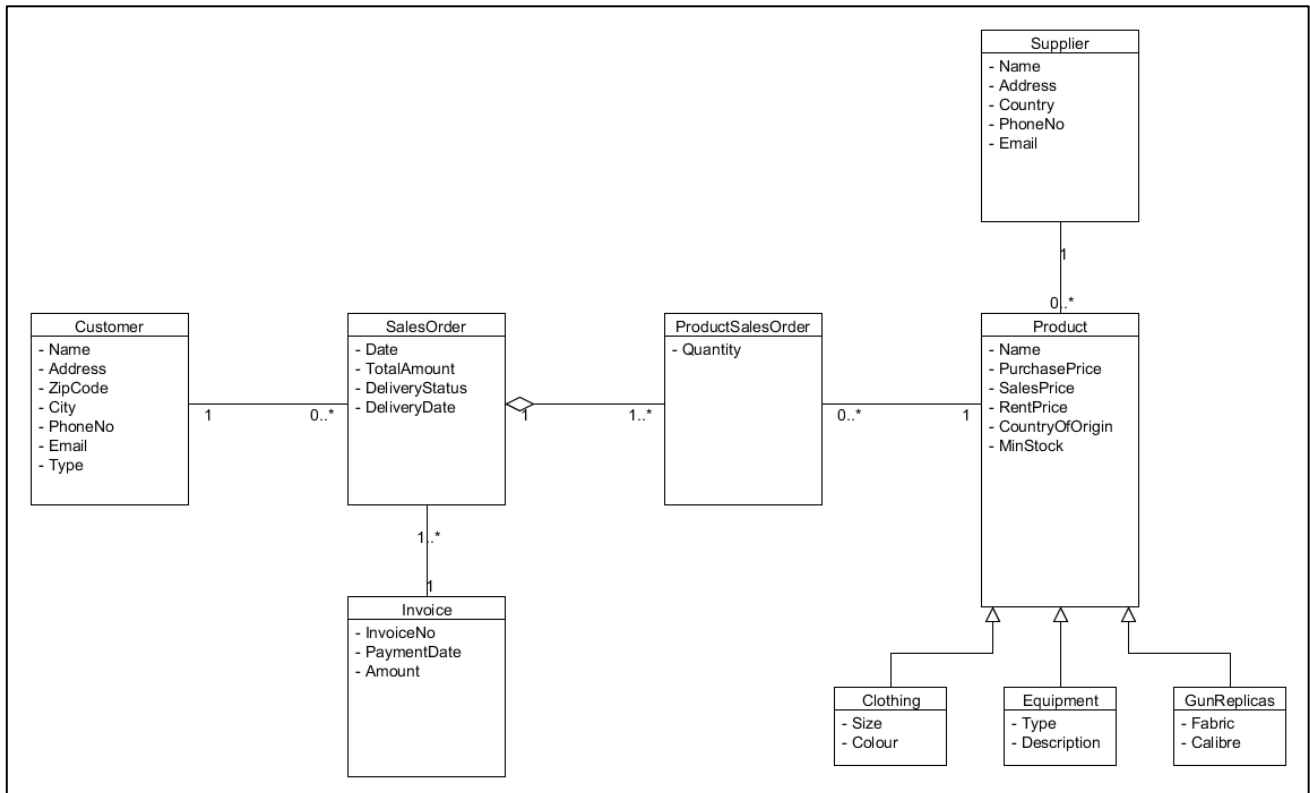
The database schema shows the structure of our database. It shows the tables and the relationship between the tables. We 7 tables namely;

- Customer, ZipcodeCity, Supplier, Invoice, SalesOrder, ProductSalesOrder, and Product.

Customer table consists of the attributes to represent a customer in our domain. It uses **custID** as its primary key. An interesting point in Customer table is the zipCode column. We normalized Customer table. This is because there is a functional dependence between *zip-code* and *city* columns. We derived a new table called zipCodeCity it uses **zipcode** as primary key; this new table stores zip codes and their corresponding city names. Customer table has a foreign key to Zipcode city table.

- Supplier table consists of attributes to represent suppliers in our domain. It uses **supplierID** as its primary key.
- Invoice table consists of attributes to represent invoice in our domain. It uses **invoiceNo** as primary key.
- SalesOrder table consists of attributes to represent orders in our domain. It uses **salesorderID** as primary key. It has a foreign key to invoiceNo and custID.
- ProductSalesOrder table is used to represent partorders. It is a new table that was added to the domain to handle partorders. This is because there is no object referencing in databases. When we have many to many, relationship in our domain, we cannot use list over our data, therefore we make a new table that joins both tables to form a new table and combine their primary keys as the primary key of the new table. ProductSalesOrder table uses the foreign keys **salesorderID** and **productID** as its primary key (super key). It also uses the column name **quantity** to represent the number of each product.
- Product table consists of attributes to represent products in our domain. It uses **productID** as primary key. It has a foreign key to supplierID.

DOMAIN MODEL



1 Det mest åbenlyse vi har ændret i den domæne model vi fik udleveret, er at vi har indsat produktSalesOrder klassen. Det har vi gjort for at kunne få antal med på de vare, der bliver bestilt, så at man ikke kun kan bestille 1 af noget og så hvis man skal bruge flere så bestille en flere gange. Vi har sat en aggregering fra produktSalesOrder klassen til salesOrder klassen da produktSalesOrder klassen bruger alle de ting der er i salesOrder klassen og så sætter den antal på hver vare og dermed kan salesOrder klassen regne TotalAmount.

SQL SCRIPTS FOR OPRETTELSE AF DATABASEN

```
1 create table Customer
2 (fname varchar(20) not null,
3 lname varchar(20) not null,
4 custid int,
5 address varchar(30) not null,
6 zipcode int,
7 city varchar(20),
8 phoneno int not null,
9 email varchar(50) not null,
10 type varchar(15) not null,
11 primary key (custid));
12
13
14
15 create table Supplier
16 (supplierid int not null,
17 name varchar (20),
18 address varchar (30),
19 country varchar (30),
20 phonenum int not null,
21 email varchar (50) not null,
22 primary key (supplierid));
23
24
25 create table Invoice
26 (invoicenum int,
27 paymentdate datetime DEFAULT GETDATE(),
28 totalamount decimal(10, 2),
29 primary key (invoicenum));
30
31
32 create table Product
33 (productid int,
34 purchaseprice decimal(10, 2),
35 salesprice decimal(10, 2),
36 rentprice decimal (10, 2),
37 countryoforigin varchar(30),
38 minimumstock int,
39 size char (9),
40 colour varchar(20),
41 producttype varchar (20),
42 productdescription varchar (50),
43 fabric varchar (50),
44 supplier int,
45 calibre decimal (4,4)
46 primary key (productid),
47 foreign key (supplier) references Supplier(supplierid));
48
49
50 create table SalesOrder
51 (salesorderid int,
52 creationdate datetime DEFAULT GETDATE(),
53 deliverydate datetime DEFAULT GETDATE(),
54 deliverystatus varchar(20),
55 totalamount decimal (10, 2),
56 customer int not null,
57 invoice int
```

```
58 primary key (salesorderid),
59 foreign key (customer) references Customer(custid),
60 foreign key (invoice) references Invoice (invoicenum));
61
62
63
64 create table ProductSalesOrder
65 (prodid int,
66 salesid int,
67 quantity int not null,
68 foreign key(prodid) references Product(productid) on update cascade on delete
69 cascade,
70 foreign key(salesid) references SalesOrder(salesorderid) on update cascade on
71 delete cascade);
```


SQL SCRIPTS FOR INDSÆTTELSE AF DATA TIL DATABASEN

Indsættelse af Products

```

1
2
3 insert into Product values (111, 20.50, 30.50, 5.50, 'USA', 50, 'large',
'blue',
null, null, null, 330, null, 'Cowboy Hat', 'Clothing');
4 insert into Product values (112, 21.50, 31.50, 4.50, 'Denmark', 50, 'medium',
'green', null, null, null, 331, null, 'Cowboy Boots', 'Clothing');
5 insert into Product values (113, 32.50, 50.50, 5.50, 'France', 50, 'medium',
'yellow', null, null, null, 332, null, 'Ponchos', 'Clothing');
6 insert into Product values (114, 50.50, 65.50, 5.50, 'Germany', 50, 'large',
'blue',
null, null, null, 333, null, 'Denims', 'Clothing');
7
8
9
10
11 insert into Product values (115, 11.50, 21.50, 4.50, 'USA', 50, null, null,
'New
Model', 'describe prod', null, 334, null, 'Licence Plates', 'Equipment');
12 insert into Product values (116, 27.50, 37.50, 5.50, 'Nigeria', 50, null,
null, 'New
Model', 'describe prod', null, 335, null, 'Bags', 'Equipment');
13 insert into Product values (117, 25.50, 41.50, 4.50, 'Faroe Islands', 50,
null,
null, 'New Model', 'describe prod', null, 336, null, 'Ornaments', 'Equipment');
14 insert into Product values (118, 20.50, 30.50, 5.50, 'Hawai', 50, null, null,
'New
Model', 'describe prod', null, 337, null, 'Pistol Belts', 'Equipment');
15
16
17
18 insert into Product values (119, 71.50, 91.50, 4.50, 'France', 50, null,
null, null,
null, 'silver', 338, 0.02, 'Texas Ranger', 'GunReplicas');
19 insert into Product values (120, 28.50, 60.50, 5.50, 'Denmark', 50, null,
null,
null, null, 'gold', 339, 0.18, 'Deringer Pistol', 'GunReplicas');
20 insert into Product values (121, 43.50, 67.50, 5.50, 'Faroe Islands', 50,
null,
null, null, null, 'metal', 340, 0.1, 'Frontier Revolver', 'GunReplicas');
21 insert into Product values (122, 61.50, 90.50, 4.50, 'Nigeria', 50, null,
null,
null, null, 'aluminium', 341, 0.4, 'Dragoon Revolver', 'GunReplicas');
22

```

Indsættelse af Supplier

```

1 insert into Supplier values (330, 'Aarhus A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
2 insert into Supplier values (331, 'Aabybro A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
3 insert into Supplier values (332, 'Hamburg A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
4 insert into Supplier values (333, 'Nike', 'Aalbogvej 13', 'USA', 50124574,

```

```
'aa@aalb.dk');
5 insert into Supplier values (334, 'Denim A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
6 insert into Supplier values (335, 'Cowboys A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
7 insert into Supplier values (336, 'Copenhagen A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
8 insert into Supplier values (337, 'Thisted A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
9 insert into Supplier values (338, 'Cromwell A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
10 insert into Supplier values (339, 'Lagoon A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
11 insert into Supplier values (340, 'UCN A/S', 'Aalbogvej 13', 'USA', 50124574,
'aa@aalb.dk');
12 insert into Supplier values (341, 'Crown A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
13 insert into Supplier values (342, 'Longoria A/S', 'Aalbogvej 13', 'USA',
50124574,
'aa@aalb.dk');
14
```

Indsættelse af Customers

```
1 insert into Customer values('Høgni', 'Juul', '444', 'Marie Curies alle',
'9220',
'Aalborg', '53501204', 'hognijuul@gmail.com', 'Kunde'
2 insert into Customer values('Jakob', 'Kaspersen', '555', 'Læsøgade', '9000',
'Aalborg', '22850582', 'jakob.kaspersen@gmail.com', 'Kunde'
3 insert into Customer values('Søren', 'vege', '666', 'Sigred Undsetsvej',
'9220',
'Aalborg', '26402498', 'mail@vegeberg.net', 'Kunde'
4 insert into Customer values('Kaj', 'Olsen', '777', 'Dannebrosgade', '9000',
'Aalborg', '61994853', 'kajnet@gmail.com', 'Kunde'
```