

# ER-modell

Databas för sveriges kraftverk- Programmeringstenta

Julia Nilsson - juni22

# Innehållsförteckning

<b>Innehållsförteckning</b>	<b>2</b>
<b>Konceptuell modell</b>	<b>2</b>
Beskriv databasen i ett textstycke.	2
Skriv ned alla entiteter.	3
Skriv ned alla relationer och visa i matris.	3
Rita enkelt ER-diagram med entiteter och relationer.	3
Komplettera ER-diagram med kardinalitet.	3
Komplettera ER-diagram med alla attribut samt kandidatnycklar.	3
<b>Logisk modell</b>	<b>3</b>
Modifiera ER-diagram enligt relationsmodellen	3
Utöka ER-diagram med primära/främmande nycklar samt kompletterande attribut	3
<b>Fysisk modell</b>	<b>3</b>
Skapa SQL DDL för tabellerna	3
Lista funktioner som databasen skall stödja(API)	3
<b>Appendix</b>	<b>5</b>
Bilaga 1- SQL DDL	5

# Konceptuell modell

## Beskriv databasen i ett textstycke.

Databasen består av kraftverk, konsumenter och typer. Den innehåller dessutom två kopplingstabeller, kraftverk och typer samt konsumenter och kraftverk.

Det finns olika kraftverk som man vill ha koll på och köpa el från. Man kan se kraftverkens elproduktion och nyttjandegrad. Det går att beräkna årsproduktionen med formeln  $\text{effekt} * 365 * \text{nyttjandegrad} / 100$ .

Konsumenterna är de som vill köpa elen. Tabellen innehåller deras information samt deras årsbehov av el.

Typ tabellen innehåller de olika typer av el som går att köpa, det är bra att se när konsumenterna bara vill köpa viss typ av el.

## Skriv ned alla entiteter.

- Customer
- Powerplant
- Type

## Skriv ned alla relationer och visa i matris.

*Customer* köper från ett/flera *Powerplant*

*Customer* väljer en *Type*

*Powerplant* förser en/flera *Customer*

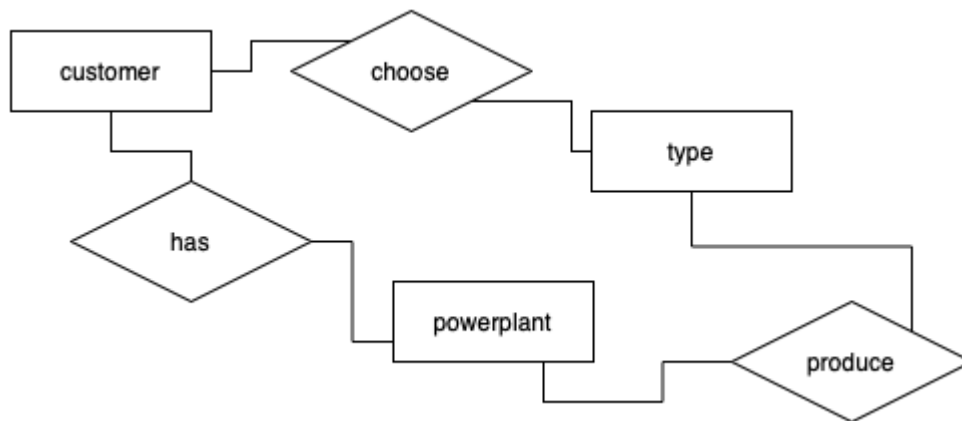
*Powerplant* producerar en/flera *Type*

*Type* producera på en/flera *Powerplant*

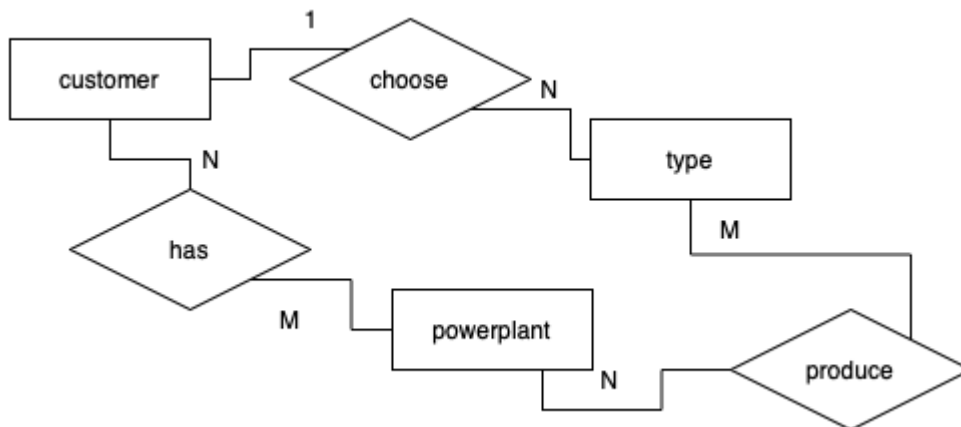
*Type* används av en/flera *Customer*

Entiteter	<b>customer</b>	<b>powerplant</b>	<b>type</b>
<b>customer</b>		köper från	väljer
<b>powerplant</b>	förser		producerar
<b>type</b>	används av	produceras	

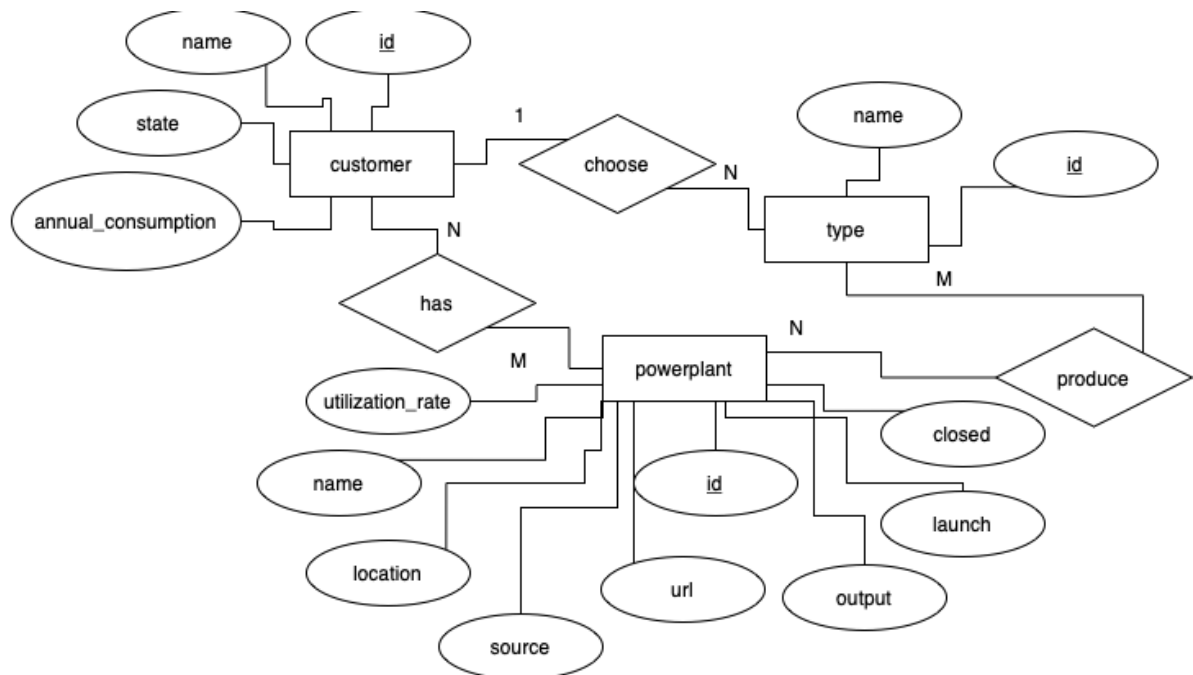
Rita enkelt ER-diagram med entiteter och relationer.



Komplettera ER-diagram med kardinalitet.

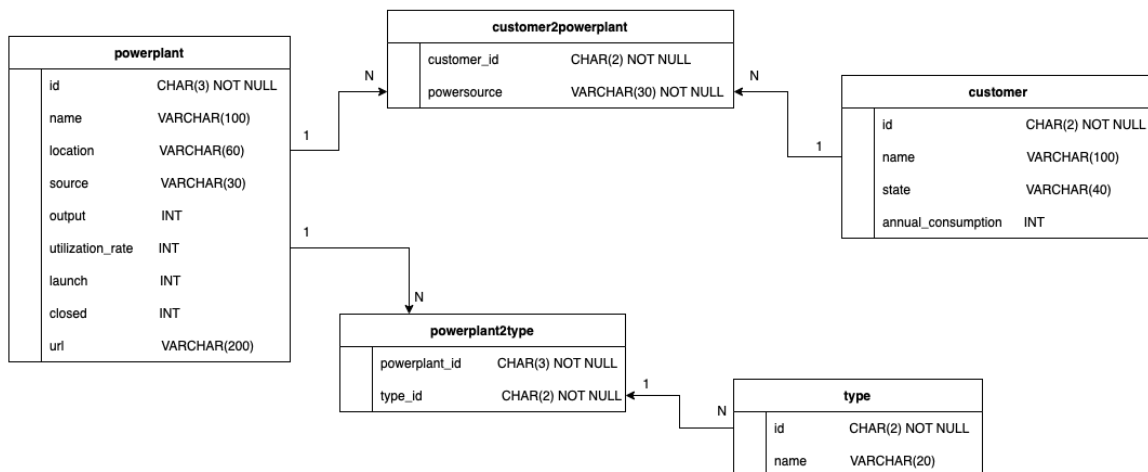


Komplettera ER-diagram med alla attribut samt kandidatnycklar.

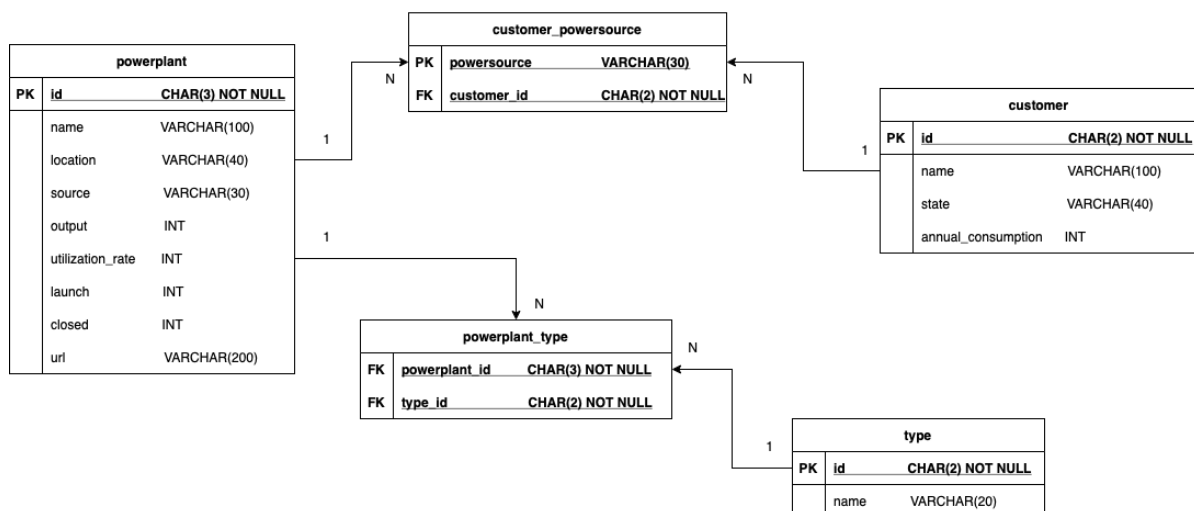


# Logisk modell

## Modifiera ER-diagram enligt relationsmodellen



## Utöka ER-diagram med primära/främmande nycklar samt kompletterande attribut



# Fysisk modell

## Skapa SQL DDL för tabellerna

Jag valde att skapa min SQL DDL för hand. Så jag skapade de bara direkt i Visual studio med stöd från min logiska modell. För resultat se Bilaga 1- SQL DDL.

## Lista funktioner som databasen skall stödja(API)

1. Visa alla kunder
2. Visa alla kraftverk
3. Visa vilka energikällor kunderna använder
4. Visa vilken typ av energi kraftverken använder

# Appendix

## Bilaga 1- SQL DDL

```
DROP TABLE IF EXISTS `customer_powersource`;  
DROP TABLE IF EXISTS `powerplant_type`;  
DROP TABLE IF EXISTS `customer`;  
DROP TABLE IF EXISTS `type`;  
DROP TABLE IF EXISTS `powerplant`;
```

```
CREATE TABLE `powerplant` (  
  `id` CHAR(3) NOT NULL,  
  `name` VARCHAR(100),  
  `location` VARCHAR(40),  
  `source` VARCHAR(30),  
  `output` INT,  
  `utilization_rate` INT,  
  `launch` INT,  
  `closed` INT,  
  `url` VARCHAR(200),  
  
  PRIMARY KEY (`id`)  
);
```

```
CREATE TABLE `type` (  
  `id` CHAR(2) NOT NULL,  
  `name` VARCHAR(20),  
  
  PRIMARY KEY (`id`)  
);
```

```
CREATE TABLE `customer` (  
  `id` CHAR(2) NOT NULL,  
  `name` VARCHAR(100),  
  `state` VARCHAR(40),  
  `annual_consumption` INT,  
  
  PRIMARY KEY (`id`)  
);
```



```
CREATE TABLE `customer_powersource` (  
  `powersource` VARCHAR(30) NOT NULL,  
  `customer_id` CHAR(2) NOT NULL,  
  
  PRIMARY KEY (`powersource`),  
  FOREIGN KEY (`customer_id`) REFERENCES `customer`(`id`)  
);  
  
CREATE TABLE `powerplant_type` (  
  `powerplant_id` CHAR(3) NOT NULL,  
  `type_id` CHAR(2) NOT NULL,  
  
  FOREIGN KEY (`powerplant_id`) REFERENCES `powerplant`(`id`),  
  FOREIGN KEY (`type_id`) REFERENCES `type`(`id`)  
);
```