	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2020/21
	Aufgabenzettel	2			
	STiNE-Gruppe 01	Coban, Giese, Kondrashova, Krüger, Oosting			
	Ausgabe	Mi. 1.11.2023	Abgabe	Do. 17.11.2023	

1

```
CREATE TABLE Person (
  PNr INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
  Vorname TEXT NOT NULL,
  Nachname TEXT NOT NULL
);
```


```
CREATE TABLE Manager (
  PNr INT UNSIGNED PRIMARY KEY,
  Bueroadresse TEXT NOT NULL,
  UNIQUE(Bueroadresse),
  FOREIGN KEY (PNr) REFERENCES Person(PNr)
);
```

```
CREATE TABLE Musiker (
  PNr INT UNSIGNED PRIMARY KEY,
  Stil TEXT NOT NULL,
  FOREIGN KEY (PNr) REFERENCES Person(PNr)
);
```

```
CREATE TABLE Produzent (
  PNr INT UNSIGNED PRIMARY KEY,
  ZahlVonPlatinSchallplatten INT UNSIGNED,
  FOREIGN KEY (PNr) REFERENCES Person(PNr)
);
```

```
CREATE TABLE Band (
  Name VARCHAR(512) PRIMARY KEY,
  Gruendung DATE NOT NULL,
  Aufloesung DATE,
  Bandleader INT UNSIGNED NOT NULL,
  CHECK (Aufloesung > Gruendung),
  FOREIGN KEY (Bandleader) REFERENCES Musiker(PNr)
);
```

```
CREATE TABLE Bandmitglied (
  Band VARCHAR(512) NOT NULL,
  Musiker INT UNSIGNED NOT NULL,
  FOREIGN KEY (Band) REFERENCES Band(Name),
```

	Lehrveranstaltung	Grundlagen von Datenbanken WS 2020/21		
	Aufgabenzettel	2		
	STiNE-Gruppe 01	Coban, Giese, Kondrashova, Krüger, Oosting		
	Ausgabe	Mi. 1.11.2023	Abgabe	Do. 17.11.2023

```
FOREIGN KEY (Musiker) REFERENCES Musiker(PNr),
PRIMARY KEY (Band, Musiker)
);
```


```
CREATE TABLE ManagerManagedBand (
Band VARCHAR(512) NOT NULL,
Manager INT UNSIGNED NOT NULL,
Gewinnbeteiligung DECIMAL NOT NULL,
FOREIGN KEY (Band) REFERENCES Band(Name),
FOREIGN KEY (Manager) REFERENCES Manager(PNr),
PRIMARY KEY (Band, Manager),
CHECK (Gewinnbeteiligung >= 5.0),
CHECK (Gewinnbeteiligung <= 50.0)
);
```

```
CREATE TABLE Produziert (
ID INT UNSIGNED PRIMARY KEY AUTO_INCREMENT
);
```

```
CREATE TABLE BandProduziert (
Produziert INT UNSIGNED NOT NULL,
Band VARCHAR(512) NOT NULL,
FOREIGN KEY (Produziert) REFERENCES Produziert(ID),
FOREIGN KEY (Band) REFERENCES Band(Name),
PRIMARY KEY (Band, Produziert)
);
```

```
CREATE TABLE ProduzentProduziert (
Produzent INT UNSIGNED NOT NULL,
Produziert INT UNSIGNED NOT NULL,
FOREIGN KEY (Produziert) REFERENCES Produziert(ID),
FOREIGN KEY (Produzent) REFERENCES Produzent(PNr),
PRIMARY KEY (Produzent, Produziert)
);
```

```
CREATE TABLE Album (
Name VARCHAR(512) PRIMARY KEY,
Releasedatum DATE NOT NULL,
Preis INT UNSIGNED NOT NULL,
NummerVonSongs INT UNSIGNED NOT NULL,
Laufzeit INT UNSIGNED NOT NULL,
```

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2020/21
	Aufgabenzettel	2			
	STiNE-Gruppe 01	Coban, Giese, Kondrashova, Krüger, Oosting			
	Ausgabe	Mi. 1.11.2023	Abgabe	Do. 17.11.2023	

```

Produziert INT UNSIGNED NOT NULL,
FOREIGN KEY (Produziert) REFERENCES Produziert(ID)
);

```

```

CREATE TABLE Song (
Name VARCHAR(512) NOT NULL,
Releasedatum DATE NOT NULL,
Laufzeit INT UNSIGNED NOT NULL,
Musikrichtung TEXT NOT NULL,
Band VARCHAR(512) NOT NULL,
FOREIGN KEY (Band) REFERENCES Band(Name),
PRIMARY KEY (Name, Band)
);

```

```

CREATE TABLE SongIstTeilVonAlbum (
Song VARCHAR(512) NOT NULL,
Band VARCHAR(512) NOT NULL,
Album VARCHAR(512) NOT NULL,
FOREIGN KEY (Song, Band) REFERENCES Song(Name, Band),
FOREIGN KEY (Album) REFERENCES Album(Name),
PRIMARY KEY (Song, Band, Album)
);

```

2


2.1

2.1.1

```

SELECT DISTINCT Figur.FNr, Figur.Name, Figur.'Alter'
FROM Figur, Comic, KommtVor
WHERE Figur.Alter >= 25
AND KommtVor.Figur = Figur.FNr
AND Comic.CNr = KommtVor.Comic
AND Comic.Autor <> 'Peter David';

```

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2020/21
	Aufgabenzettel	2			
	STiNE-Gruppe 01	Coban, Giese, Kondrashova, Krüger, Oosting			
	Ausgabe	Mi. 1.11.2023	Abgabe	Do. 17.11.2023	

2.1.2

```
SELECT DISTINCT Figur.FNr, Figur.Name, Figur.'Alter'
FROM Figur, Comic, KommtVor
WHERE Figur.Alter >= 25
AND KommtVor.Figur = Figur.FNr
AND Comic.CNr = KommtVor.Comic
AND Comic.Autor <> 'Peter David';
```

2.1.3

```
SELECT DISTINCT Verlag.VNr, Verlag.Verlagsname
FROM Verlag, SuperheldIn, Figur
WHERE Verlag.VNr = Figur.Besitzer
AND Figur.FNr = SuperheldIn.SNr
GROUP BY Verlag.VNr, Verlag.Verlagsname
HAVING COUNT (SNr) >= 30
AND AVG(Figur.'Alter');
```

2.2

```
SELECT DISTINCT Verlag.VNr, Verlag.Verlagsname
FROM Verlag, SuperheldIn, Figur, Comic
WHERE Verlag.MostIconicHero = SuperheldIn.SNr
AND SuperheldIn.SNr = Figur.FNr
AND Verlag.MostIconicHero = SuperheldIn.SNr
AND Comic.Preis < 20
AND Comic.Autor LIKE 'Alan%';
```