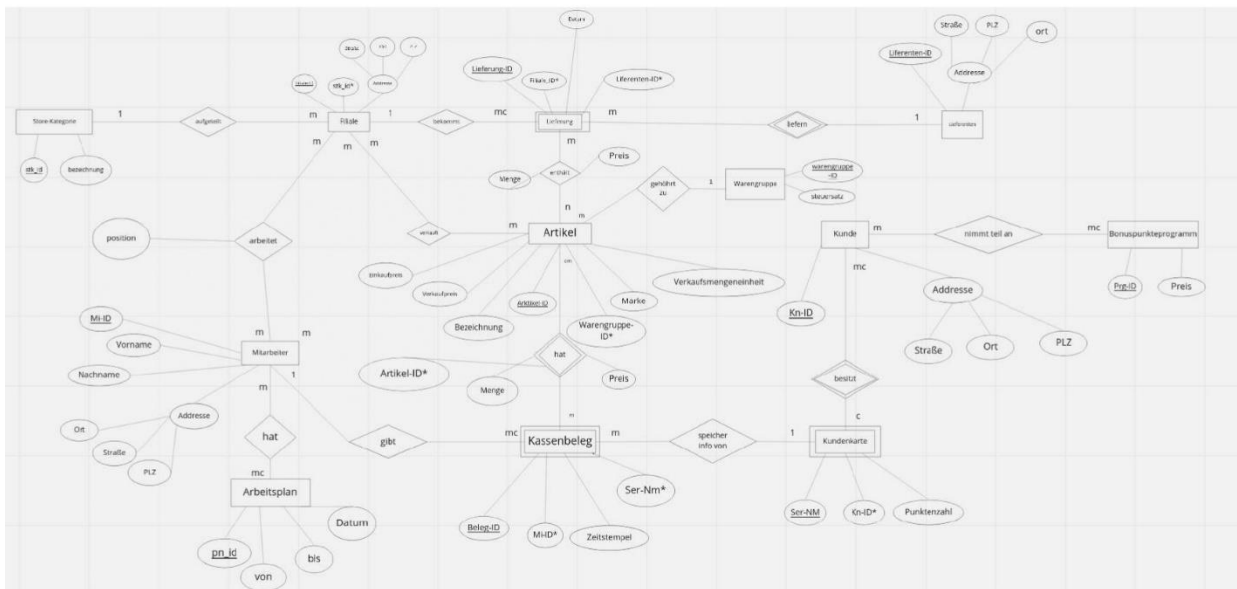


Dokumentation

1: Einzelhandel

1: Das semantische Datenmodell



2: Die Erklärungen einzelner Relationen

- Die Filiale enthält ein Angebot von Lieferanten, das aus mehreren Artikeln besteht. Die Artikel werden in verschiedene Warengruppen eingeteilt und verkauft.
- Die Mitarbeiter haben unterschiedliche Positionen und Arbeitszeiten. Als Kassierer muss man dem Kunden den Kassenbon aushändigen, der für die Speicherung der Informationen in der Kundenkarte benötigt wird.

- Der Kunde kann mit der Anzahl der Punkte an einem Bonuspunkteprogramm teilnehmen und einen Preis gewinnen.

3: Die SQL-Abfragen

3.1

--Die Mitarbeiter, die als Leiter arbeiten

```
SELECT m.mr_id, m.vorname, m.nachname, m.strasse, m.ort, m.plz FROM mitarbeiter m
```

```
JOIN arbeitet ar ON ar.mr_id = m.mr_id
```

```
WHERE ar.position = 'Leiter';
```

Data Output							Explain	Messages	Notifications
	mr_id [PK] integer	vorname character varying (30)	nachname character varying (30)	strasse character varying (50)	ort character varying (50)	plz character (5)			
1	1	Chandler	Fossett	Scottstr	Giessen	[null]			
2	4	Colette	Dy	Crossingstr	Leipiz	[null]			
3	7	Bill	Aisman	Rosenstr	Stuttgart	77120			

3.2

--Mitarbeiter die keinen Arbeitsplan haben

```
SELECT mr.mr_id, mr.vorname, mr.nachname, mr.strasse, mr.ort, mr.plz FROM mitarbeiter mr
```

```
WHERE mr.mr_id IN(
```

```
SELECT m.mr_id FROM mitarbeiter m
```

```
EXCEPT
```

```
SELECT mhap.mr_id FROM mitarbeiter_hat_arbeitsplan mhap);
```

Data Output		Explain	Messages	Notifications		
	<div>mr_id</div> <div>[PK] integer</div>	<div>vorname</div> <div>character varying (30)</div>	<div>nachname</div> <div>character varying (30)</div>	<div>strasse</div> <div>character varying (50)</div>	<div>ort</div> <div>character varying (50)</div>	<div>plz</div> <div>character (5)</div>
1	3	Alessandro	Puleston	sylvaniastr	Potsdam	[null]
2	4	Colette	Dy	Crossingstr	Leipiz	[null]
3	2	Benedicto	Pittham	Valleystr	Kiel	89500

3.3

--Azahl der Teilnehmen am Bonuspunteprogramm(jeweils)

```
SELECT COUNT(t.kn_id),t.prog_id FROM tilnahme_bonusprogramm t
GROUP BY t.prog_id;
```

	Data Output	Explain	Messages	Notifications
	count bigint	prog_id integer		
1	1	3		
2	2	5		
3	1	4		
4	1	2		
5	2	1		

3.4

--Die Artikel, die in der Lieferung (ID = 1) enthalten








```
SELECT ark.bezeichnung FROM artikel ark
WHERE ark.ark_id IN (
SELECT en.ark_id FROM enthaelt en
JOIN artikel ar ON en.ark_id = ar.ark_id
WHERE en.lieferung_id = 1);
```

	Data Output	Explain	Messages	Notifications
	bezeichnung character varying (50)			
1	Soup Campbells			
2	Longos - Chicken Cordon Bleu			

3.5

--- Alle Mitarbeiter, die mit „B“ beginnen.

```
SELECT mr.mr_id, mr.vorname, mr.nachname, mr.strasse, mr.ort, mr.plz FROM mitarbeiter mr
WHERE mr.vorname LIKE 'B%';
```

Data Output		Explain	Messages	Notifications		
	mr_id [PK] integer 	vorname character varying (30) 	nachname character varying (30) 	strasse character varying (50) 	ort character varying (50) 	plz character (5) 
1	2	Benedicto	Pittham	Valleystr	Kiel	89500
2	7	Bill	Aisman	Rosenstr	Stuttgart	77120
3	10	Bunny	Jiggen	Maxstr	Kaiserslauten	[null]

3.6

--Der Kunde mit der höchsten Punktezahl

```
SELECT kn.kn_id, kn.vorname, kn.nachname, kn.strasse, kn.ort, kn.plz FROM kunde kn
WHERE kn.kn_id = ( SELECT knk.kn_id FROM kundenkarte knk
WHERE knk.punkenzahl = (SELECT MAX(knr2.punkenzahl) FROM kundenkarte knr2)) ;
```

Data Output

Explain

Messages

Notifications

	kn_id [PK] integer	vorname character varying (30)	nachname character varying (30)	strasse character varying (50)	ort character varying (50)	plz character (5)
1	1	Debby	Bartke	Brickstr	Heidelberg	[null]

3.7

--Anzahl der 'Hyper-Super-Mega-Store' der Filialen

```
SELECT COUNT(f.filiale_id)FROM filiale f
JOIN storekategorie stk ON f.stk_id = stk.stk_id
WHERE stk.bezeichnung = 'Hyper-Super-Mega-Store';
```

Data Output	Explain	Messages	Notifications
	count bigint		
1	5		

3.8

--Filialen mit plz zwischen 2000 und 3000, aufsteigen sortiert

```
SELECT fl.filiale_id, fl.strasse, fl.ort, fl.plz FROM filiale fl
WHERE fl.plz BETWEEN '20000' and '30000'
ORDER BY fl.ort ASC;
```

Data Output		Explain	Messages	Notifications
	filiale_id [PK] integer	strasse character varying (50)	ort character varying (50)	plz character (5)
1	5	Lyonsstr	Kassel	22011
2	4	Fultonstr	Marburg	28570

4: Selbstkritische Diskussion

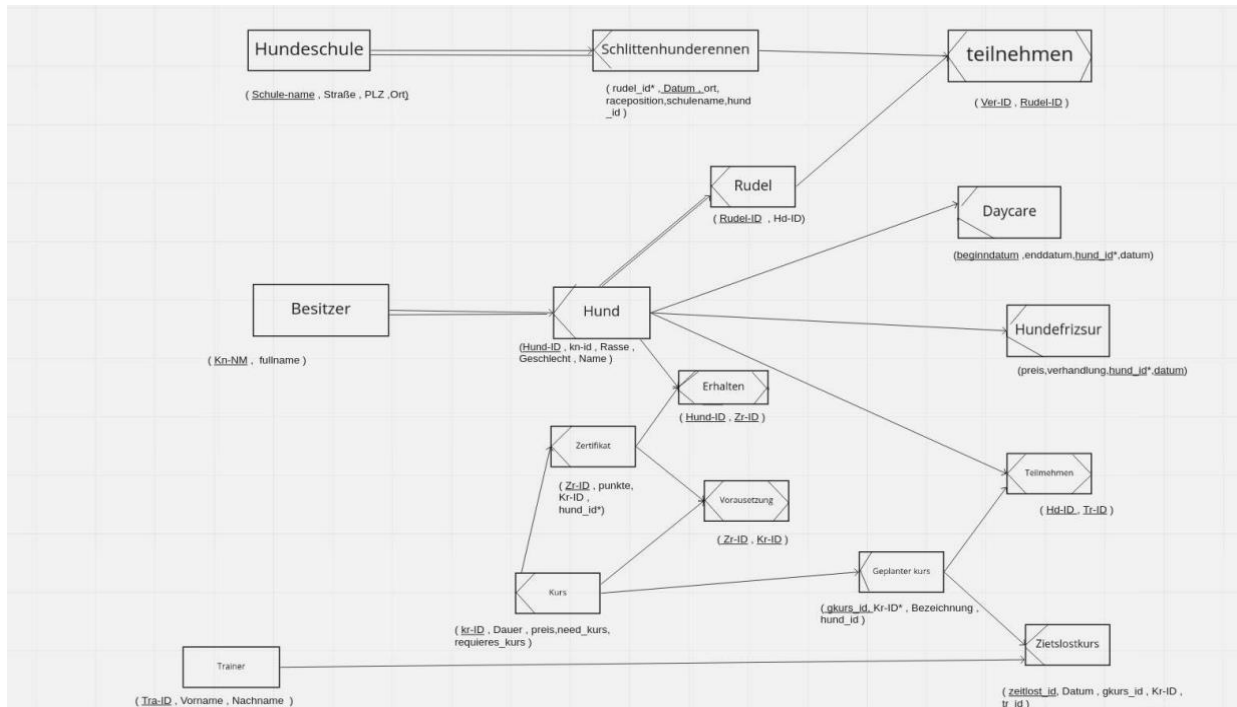
Es ist nicht möglich, direkt auf den bezahlten Beitrag zuzugreifen, sondern mit Hilfe der Beziehungstabelle zwischen dem Artikel und dem Beleg.

5: Beschreibung und Erklärung der Erweiterung.

Als 1. Erweiterung habe die die Tabelle Kassenbeleg angelegt , die bei der Speicherung von Daten auf der Kundenkarte hilft und die 2.Erweiterung ist die Tabelle Arbeitsplan , die es dem Filialleiter ermöglicht, die Arbeitszeiten einfach zu planen.

2: Hundeschule

2.1: Das semantische Datenmodell



2.2: Die Erklärungen einzelner Relationen

- Der Besitzer hat Hund(e) und jeder Hund hat verschiedene Aktivitäten, der könnte in einem Race(Schlittenhunderennen) durch die Teilnahme in einem Rudel.
- Der Hund kann in: Geplantekurs teilnehmen wenn sie die Voraussetzungen erhalten haben ||, oder in der Daycare oder Hunderfrisur gehen.
- Der Hund kann ein Zertifikat von den Geplantenkursen, dass er bestanden hat bekommen.
- Die Hundeschule organisiert die Race und geben die Preise und die Zeit die jeder Rudel hatte in einem bestimmte Schlittenhunderennen.
- Jeder Geplantekurs hat ein Zeitslot (liste die Termine) und die Trainer Id dass diesem Termin betreut.

2.3: Die SQL-Abfragen

2.3.1

-- Welcher Hund hat die meisten Zertifikate gesammelt?

```
CREATE VIEW samedog(name,hund_id) AS SELECT h.name,h.hund_id ,count(g1.hund_id) AS  
"repeathund" FROM hunde h inner JOIN geplankurs g1 ON h.hund_id=g1.hund_id AND  
g1.hund_id=g1.hund_id GROUP BY h.name , h.hund_id;  
SELECT samedog.name, samedog.hund_id  
FROM samedog where "repeathund" =(select MAX ("repeathund")FROM samedog) ;
```

	Data Output	Explain	Messages	Notifications
	<div>name</div> <div>character varying (30)</div>		<div>hund_id</div> <div>integer</div>	
1	Rozalie		6	

2.3.2

--Alle hunde die schon ein kurs erfolgreich abgeschlossen haben und wie viel punkte hatte

```
SELECT DISTINCT hunde.name, hunde.rasse , hund_id,zertifikat.punkte,  
zertifikat.kurs_id,kurs.kursname FROM hunde inner join zertifikat USING (hund_id) INNER JOIN  
KURS using(kurs_id) GROUP BY hunde.name, hunde.rasse , hund_id, zertifikat.punkte,  
zertifikat.kurs_id,kurs.kursname ;
```

Data Output		Explain	Messages	Notifications		
	name character varying (30)	rasse character varying (30)	hund_id integer	punkte integer	kurs_id integer	kursname character varying (50)
1	Benedick	Husky	4	97	2	basic training 2
2	Cally	Husky	1	83	5	bodyguard dog 1
3	Cally	Husky	1	93	3	tricks 1
4	Benedick	Husky	4	96	1	basics training 1
5	Benedick	Husky	4	63	4	triks 2
6	Jacquenette	pitbull	3	76	5	bodyguard dog 1
7	Jacquenette	pitbull	3	96	3	tricks 1
8	Jacquenette	pitbull	3	57	4	triks 2
9	Cally	Husky	1	77	1	basics training 1
10	Goldarina	Boxer	2	51	2	basic training 2
11	Goldarina	Boxer	2	78	3	tricks 1
12	Benedick	Husky	4	51	3	tricks 1
13	Lily	Husky	5	65	5	bodyguard dog 1
14	Cally	Husky	1	88	1	basics training 1
15	Cally	Husky	1	96	1	basics training 1
16	Goldarina	Boxer	2	57	2	basic training 2
17	Cally	Husky	1	53	4	triks 2

2.3.3

--welche kurs haben requirites und welche kurs sind die voraussetzungen

```
SELECT kurs_id,kursname,requieres_kurs,preis FROM kurs WHERE need_kurs = true;
```

	Data Output	Explain	Messages	Notifications
	kurs_id [PK] integer	kursname character varying (50)	requieres_kurs character varying (50)	preis numeric (5,2)
1	2	basic training 2	basics training 1	300.00
2	4	triks 2	triks 1	500.00

2.3.4

--Anzahl die Hunde die in keine kurs teilnehmen.

```
SELECT count(hund_id) FROM hunde WHERE hund_id NOT IN (SELECT DISTINCT hund_id
FROM geplantekurs)
```


Data Output	Explain	Messages	Notifications
	count bigint		
1	12		

2.3.5

--Zu welchem Kurs sind durchschnittlich die meisten Doggen angemeldet?

```
CREATE VIEW anzahl_hund(gkurs_id,kurs_id,hund_id,Nr_hund) AS SELECT gkurs_id
,kurs_id,count (hund_id) AS "Nr_hund" FROM geplankkurs GROUP BY gkurs_id ,kurs_id;
SELECT a.gkurs_id, k.kursname FROM anzahl_hund a JOIN kurs k USING (kurs_id) WHERE
a.Nr_hund IN (SELECT MAX(Nr_hund) FROM anzahl_hund);
```

Data Output	Explain	Messages	Notifications
	kursname character varying (50)		
1	triks 2		

2.3.6

--Welches Rudel hat die Bestzeit im letzten Schlittenhunderennen erreicht und die hunde name und rasse ?







```
SELECT rudel_id,datum,zeit,hunde.rasse,hunde.name from schlittenhunderennen INNER JOIN
rudel USING(rudel_id) INNER JOIN hunde USING (hund_id)
WHERE
schlittenhunderennen.datum=(SELECT MAX(DATUM) FROM schlittenhunderennen ) AND
schlittenhunderennen.raceposition=1
GROUP BY
schlittenhunderennen.rudel_id,schlittenhunderennen.datum,hunde.rasse,hunde. name
```

Data Output		Explain	Messages	Notifications			
 ruel_id integer	 datum date	 zeit time without time zone	 rasse character varying (30)	 name character varying (30)	 fullname character varying (40)	 kn_id integer	
1	3	2022-05-11	01:06:10	husky	Kathe	Ginnie Woodfield	6
2	3	2022-05-11	01:06:10	husky	Zabrina	Jessalyn Vogl	4
3	3	2022-05-11	01:06:10	pitbull	Burr	Jessalyn Vogl	4

2.3.7

--fullname UND kn_id, kurs_id ,kursname ,k_pries von besitzer von jedem kurs die schon eingekauft wurde.

```
SELECT distinct h.kn_id , b.fullname, g.kurs_id ,k.kursname ,k.preis from besitzer b
inner join hunde h on b.kn_id=h.kn_id
inner join geplankkurs g on g.hund_id=h.hund_id
inner join kurs k on k.kurs_id = g.kurs_id
GROUP BY h.kn_id , b.fullname, g.kurs_id,k.kursname ,k.preis;
```

Data Output		Explain	Messages	Notifications					
 kn_id	 integer	fullname	 character varying (40)	kurs_id	 integer	kursname	 character varying (50)	preis	 numeric (5,2)
1	1	Simonne Gooderham		1	basics training 1		435.00		
2	1	Simonne Gooderham		2	basic training 2		300.00		
3	1	Simonne Gooderham		3	tricks 1		500.00		
4	1	Simonne Gooderham		4	triks 2		500.00		
5	2	Seward Feeny		1	basics training 1		435.00		
6	2	Seward Feeny		2	basic training 2		300.00		
7	2	Seward Feeny		3	tricks 1		500.00		
8	2	Seward Feeny		4	triks 2		500.00		
9	3	Sheela Aspinal		1	basics training 1		435.00		
10	3	Sheela Aspinal		2	basic training 2		300.00		
11	3	Sheela Aspinal		3	tricks 1		500.00		
12	3	Sheela Aspinal		4	triks 2		500.00		
13	4	Jessalyn vogl		2	basic training 2		300.00		
14	5	Merilee Ianni		3	tricks 1		500.00		
15	5	Merilee Ianni		4	triks 2		500.00		

2.3.8

--name,rasse alle Hunde die in keine kurs teilnehmen und ihren Besitzern .

```
SELECT besitzer.kn_id,besitzer.fullname,hunde.name,hunde.rasse FROM hunde
INNER JOIN besitzer on hunde.kn_id=besitzer.kn_id WHERE hund_id NOT IN
(SELECT DISTINCT hund_id FROM geplantekurs)

GROUP BY besitzer.kn_id ,besitzer.fullname,hunde.name ,hunde.rasse

ORDER BY fullname ASC;
```

Data Output					Explain	Messages	Notifications
	kn_id integer		fullname character varying (40)		name character varying (30)		rasse character varying (30)
1	6	Ginnie Woodfield	Gaye	Husky			
2	6	Ginnie Woodfield	Kathe	husky			
3	4	Jessalyn Vogl	Zabrina	husky			
4	4	Jessalyn Vogl	Burr	pitbull			
5	13	Melinde Davidge	Sterne	Husky			
6	13	Melinde Davidge	Melva	Boxer			
7	9	Meridel Sabberton	Leona	Chihuahua			
8	9	Meridel Sabberton	Etienne	Boxer			
9	5	Merilee Ianni	Sapphira	Husky			
10	12	Torrin Standage	Gaelan	Husky			
11	10	Zolly Scranedge	Kelbee	Chihuahua			
12	10	Zolly Scranedge	Gare	Chihuahua			

2. 4: Selbstkritische Diskussion

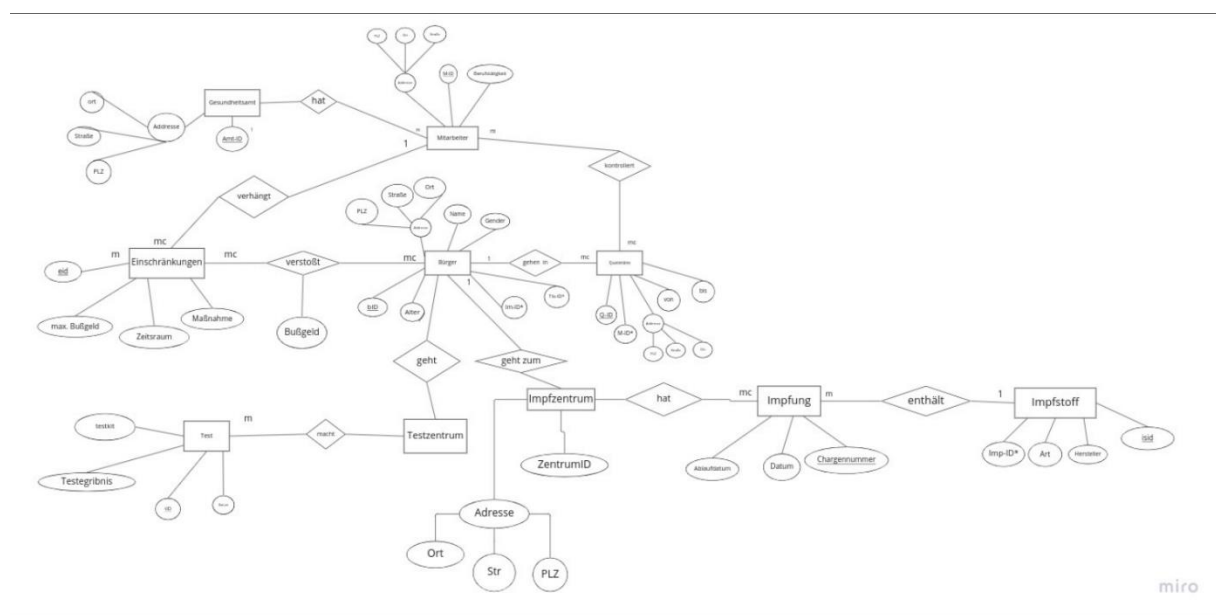
In dem Schwachen teil gibt es 2 probleme. Erstens ,dadurch dass den gleichen Kurs in verschiedene Monate angeboten wird.Außerdem könnte einen Kurs von verscheidenen Trainern angeboten werden mit dem selben ID und Kursname. Deswegen müssten wir diese Situation mit 3 identitäten Modellieren . einmal der Kurs entität mit name, kosten,kursname ,Id und dann ein Geplantekurs das ein kurs id , Beginndatum , Enddatum, Trainer_id ,Hund_id hat . Letztlich hatten wir ein Zeitslot wo jeder termin von einem Kurs gespeichert würde . Andere wichtiger Entscheidung war , wie werden wir die Voraussetzungen für jedem Kurs modellieren .Weil man das vorher überprüfen soll, ob den Hund schon den Kurs bestanden hat. Wir haben das gelöscht ,indem die Kurse haben ein boolean type variable und ein Liste von Voraussetzungen als Varchar codiert haben. Die mitarbeiter muss es überprüfen in dem man mit dem Hund_id bestätig, dass die Hunde schon die notwendige Kurse bestanden hat.

2.5: Beschreibung und Erklärung der Erweiterung.

Wir haben 2 erweiterung gemacht. Die erste Erweiterung ist die Hundefrisur da werden die Nagels , oder die haare geschnitte als auch die hunde werden gewaschen . Das könnte ein gutes kompliment für das Model weil viele Hundeschule auch ein Pet Salon anbieten , die zweite Erweiterung war die Daycare Dienst, wo man die hunde für eine gewissen Zeit lassen kann , wenn man reisen will zum Beispiel. Diese 2 Service können wichtiger für die Hundeschule sein , weil die Daten für die Verbesserung dieses Geschäft deinen kann .Wie Angeboten für die meisten gebuchte Service bei daycare ,oder für die Holidays wo die meisten Kunden daycare buchen mehr Personal für die Hundeüberwachung mieten

3: Gesundheitsamt

3.1: Das semantische Datenmodell



3.2: Die Erklärungen einzelner Relationen und Erweiterungen

- **Gesundheitsamt** , **hat Mitarbeiter** (m - n Relation) die Aufgaben der Mitarbeiter sind z.B. **Quarantäne Kontrolle**, (m - nc) und verhängt die Einschränkungen(1 - mc).
- Der **Bürger geht in Quarantäne** (m - n) wenn er/sie Covid hat und er/sie muss an der Einschränkungen halten (mc - nc) ,sonst der Mitarbeiter verhängt Bußgeld im Fall von Verstoß.

er Bürger geht zum **Testzentrum** (m - n) und macht ein Test (m - n), und/oder geht zum **Impfzentrum** (m - n) und lässt sich impfen , Impfung und Impfstoff sind Entitäten (m - 1).

- Da der Bürger geht zum **Impfzentrum** und **Testzentrum** (in der Realität) haben wir Zwei Entitäten als Erweiterung vorgeschlagen ,wir speichern die Daten in Tabellen und erstellen die zugehörigen Relationen z.B. (**Bürger geht zum Impfzentrum**), (**Bürger geht Testzentrum**)&(**Testzentrum macht Test**)

Fettgedruckt: Entität , **Rot: R-Typ** , **Blau: Erweiterung**

3.3: Die SQL-Abfragen

3.3.1

The screenshot shows a SQL query editor interface. The query is as follows:

```
1 --1. Die Anzahl der Menschen ,die Quarantäne im bestimmten Ort sind.
2 select count(bid)
3 from buerger b
4 join brgr_geht_in_quarntn using(bid)
5 join quarantaene q using(qid)
6 where q.ort = 'La Soledad';
7
```

Below the query editor, the 'Data Output' tab is active, showing the result of the query:

	count bigint
1	4

3.3.2

Gesundheitsamt/admin@dbms

Query Editor Query History

```
1 --2. in welcher Orte gab es Einschränkungen Verstöße.
2 select b.ort
3 from einschraenung e
4 join brgr_verstosst_einschrnkng bve using(eid)
5 join buerger b using(bid)
6 where e.bussgeld >0
7 GROUP BY b.ort;
```

Data Output Explain Messages Notifications

	ort
	character varying (50)
1	Qincheng
2	Huangxikou
3	Ivatsevichy
4	Birnin Kebbi
5	Mulyosari
6	Pop

3.3.3

Operators
Procedures
1.3 Sequences
Tables (20)
brgr_geht_in_quarntn
brgr_ghf_impfztrm
brgr_ghf_testztrm
brgr_verstosst_einsch
buerger
einschraenung
gesundheitsamt
gha_hat_mit
impfng_enthaelt_impf
impfstoff
impfung
impfzentrum
iz_hat_impf
kontrolle
mitarbeiter
mitarbtr_verhaengt_ei
quarantaene
test

Gesundheitsamt/admin@dbms

Query Editor Query History

```
1 select art,numofimpf
2 from numofimpfstoff n
3 where numofimpf=(select max(numofimpf) from numofimpfstoff)
4 group by art,numofimpf;
```

Data Output Explain Messages Notifications

	art	numofimpf
	character varying (50)	bigint
1	AstraZeneca	7
2	BioNTech/Pfizer	7

3.3.4

Procedures
1.3 Sequences
Tables (17)
brgr_geht_in_quarntn
brgr_verstosst_einsch
buerger
einschraenung
gesundheitsamt
gha_hat_mit
impfng_enthaelt_impf
impfstoff
impfung
impfzentrum
iz_hat_impf
kontrolle
mitarbeiter
quarantaene
test
testzentrum
tz_macht_test
Trigger Functions
Types

Gesundheitsamt/admin@dbs

Query Editor Query History

```

1 --4. Welche Mitarbeiter welche Bussgeld verhaengt hat?
2
3 select m.mid,m.ort,m.plz,e.massnahme,e.bussgeld
4 from mitarbeiter m
5 join mitarbtr_verhaengt_einschrnkng mve on mve.mid = m.mid
6 join einschraenung e ON e.eid = mve.eid
7 where e.bussgeld > 0
8 group by m.mid,e.massnahme,e.bussgeld;

```

Data Output Explain Messages Notifications

	mid	ort	plz	massnahme	bussgeld
	Integer	character varying (50)	character (5)	character varying (50)	numeric (6,2)
1	15	Saint John	53	i8f5b5a0	8649.00
2	20	Longotea	5464	i8f5b5a0	8649.00
3	18	Marxog	934	x9k7s5p6	3590.00
4	15	Saint John	53	k1p5p3n3	4686.00
5	15	Saint John	53	x9i2v8v0	8434.00

3.3.5

Gesundheitsamt/admin@dbs

Query Editor Query History

```

1 --5. welche Patient welche Impfung bekommen hat?
2
3 select b.bid,b.name,iz.izid,ib.art from buerger b
4 join brgr_ghz_impfztrm biz ON biz.bid = b.bid --Buerger geht in Impfzentrum on bid
5 join impfzentrum iz ON iz.izid = biz.izid -- on izid
6 join iz_hat_impf ihi ON ihi.izid = iz.izid --Impfzentrum hat Impfung -- on izid
7 join impfung i on ihi.chargennummer=i.chargennummer -- on chargennummer
8 join impfng_enthaelt_impfstff iei on i.chargennummer = iei.chargennummer -- on chargennummer
9 join impfstoff ib on iei.isid = ib.isid; --on isid

```

Data Output Explain Messages Notifications

	bid	name	izid	art
	Integer	character varying (50)	Integer	character varying (50)
1	2	Minette	18	AstraZeneca
2	3	Aileen	18	AstraZeneca
3	5	Astrid	18	AstraZeneca
4	3	Aileen	18	AstraZeneca
5	20	Neill	17	BioNTech/Pfizer
6	13	Costa	17	BioNTech/Pfizer
7	20	Neill	17	BioNTech/Pfizer
8	13	Costa	17	BioNTech/Pfizer

3.3.6

Query Editor Query History

```
1 --6. Risiko Patienten , die Corona haben in bestimmter Stadt
2 select b.name,count(b.bid)
3 from buerger b
4 join brgr_geht_in_quarntn bq ON bq.bid = b.bid
5 join quarantaene q ON q.qid = bq.qid
6 where b.age>=60 and q.ort in(
7     select q.ort
8     from quarantaene q
9     where ort='Marburg'
10 )
11 group by b.name;
12
```

Data Output Explain Messages Notifications

	name character varying (50)	count bigint
1	Aileen	1

3.3.7

Query Editor Query History

```
1 --7. Anzahl der Einschr ngen per Mitarbeiter
2
3 select m.mid,m.strasse,count(m.mid)
4 from einschr nung e
5 join mitarbtr_verhaengt_einschr ng mve ON mve.eid = e.eid
6 join mitarbeiter m on mve.mid = m.mid
7 group by m.mid;
8
```

Data Output Explain Messages Notifications

	mid [PK] integer	strasse character varying (50)	count bigint
1	15	Marcy	4
2	19	8th	2
3	3	Talmdage	3
4	4	Comanche	1
5	6	Maryland	1
6	13	Hollow Ridge	1

3.3.8

Gesundheitsamt/admin@dbs

Query Editor Query History

```
1 --8. Welcher Impfstoff wurde am h ufigsten genutzt?
2
3 create view number_of_impf (art , nr_impfung) as
4 select a.art,count(a.art) as "nr_impfung"
5 from impfstoff a
6 group by a.art;
7 -----
8 select art,nr_impfung
9 from number_of_impf
10 where nr_impfung=(select max(nr_impfung) from number_of_impf)
11 group by art,nr_impfung;
```

Data Output Explain Messages Notifications

	art character varying (100)	nr_impfung bigint
1	Lithium Carbonate	3
2	Moderna	3

