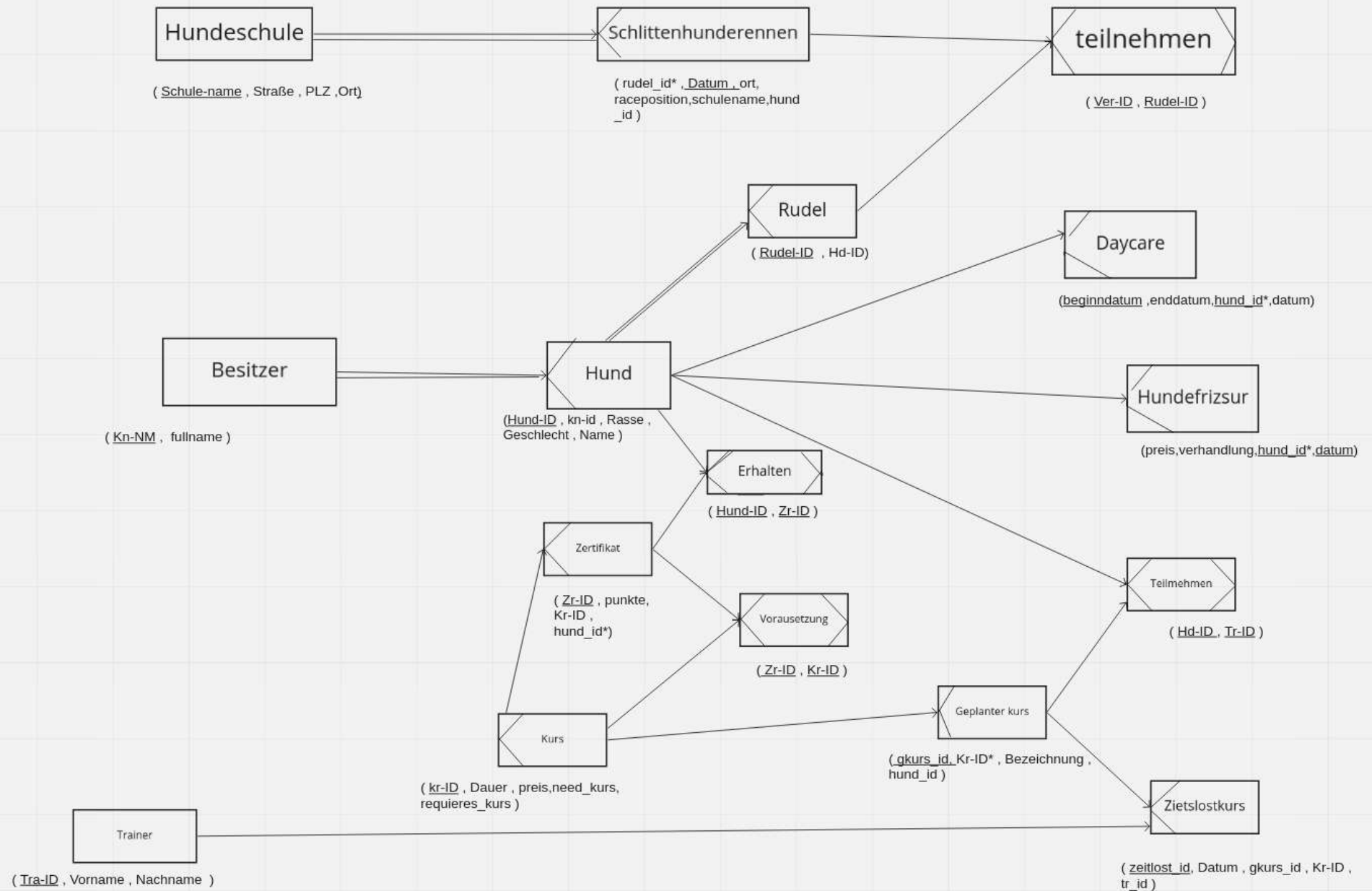


1. • Das semantische Datenmodell als eingebettetes Bild.



2•Erklärungen einzelner Relationen und Designentscheidungen.

Bei der **Daten modell** hatten wir uns für das SERM modell entschieden. Weil dadurch könnte man das Modell von links nach rechts in bezug auf die Existenzabhängigkeiten anordnen. Das half uns schneller und effektiver die wichtige Teile von dem Model zu erkennen und. Außerdem half uns zu wissen , was wir zuerst in dem Datenbank hinzufügen sollten .

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.

3.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.

4.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.

5. Die SQL-Abfragen mit Screenshots der Ergebnisse auf Ihrer Datenbank

--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) ;
```

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 ;
```

--3--welche kurs haben requisites und welche kurs sind die voraussetzungen

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

--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 geplankurs)
```

--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 geplantekurs 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);
```

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
```

--7 fullname UND kn_id, kurs_id ,kursename ,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 geplantekurs 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;
```

--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;
```

pgAdmin File Object Tools Help admin@admin.com (internal)

Browser Dashboard Properties SQL hunderschule/admin@dbs

Servers (1)

- dbs
 - Databases (6)
 - Geschäftsreisen
 - Projektmanagement
 - Videothekenkette
 - dbs
 - hunderschule**
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (13)
 - adresse

Query Editor Query History

```
1 --1 Welcher Hund hat die meisten Zertifikate gesammelt?
2 --CREATE VIEW samedog(name,hund_id) AS
3 --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
4 --g1.hund_id=g1.hund_id GROUP BY h.name , h.hund_id;
5
6
7 SELECT samedog.name, samedog.hund_id FROM samedog where "repeathund" =(select MAX ("repeathund")FROM samedog) ;
8
9
10
```

Data Output Explain Messages Notifications

	name	hund_id
	character varying (30)	integer
1	Rozalie	6


```
1
2 --2 Alle Kurse die erfolgreich abgeschlossen wurden , name ,rasse de und wie viel punkte hatte
3
4 SELECT DISTINCT hunde.name, hunde.rasse , hund_id,zertifikat.punkte, zertifikat.kurs_id,kurs.kursname
5 FROM hunde INNER JOIN zertifikat USING (hund_id)
6 INNER JOIN KURS using(kurs_id)
7 GROUP BY hunde.name, hunde.rasse , hund_id, zertifikat.punkte, zertifikat.kurs_id,kurs.kursname ;
8
9
10
```

	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

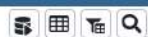
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- ▼ Schemas (1)
 - ▼ public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators

```

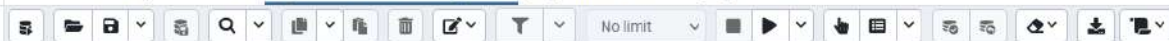
1
2 --3--Select name von kurs, kurs_id m preis und name die voraussetzungen kurs von den Kursen die andere Kurs als voraussetzung haben
3 SELECT kurs_id,kursname,requieres_kurs,preis FROM kurs WHERE need_kurs = true;
    
```

	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

Browser



Dashboard Properties SQL **hunderschule/admin@dbs** public.hunde/h... public.besitzer...



hunderschule/admin@dbs

Query Editor Query History

```
1
2  --4 - - --Anzahl die Hunde die in keine kurs teilnehmen;
3  SELECT count(hund_id) FROM hunde WHERE hund_id NOT IN (SELECT DISTINCT hund_id FROM geplanteKurs)
4
5
```

Data Output Explain Messages Notifications

	count	bigint
1		12

- > Geschäftsreisen
- > Projektmanagement
- > Videothekenkette
- > dbs
- ✓ hunderschule
 - > Casts
 - > Catalogs
 - > Event Triggers
 - > Extensions
 - > Foreign Data Wrappers
 - > Languages
 - > Publications
 - ✓ Schemas (1)
 - ✓ public
 - > Aggregates
 - > Collations






- > Geschäftsreisen
- > Projektmanagement
- > Videothekenkette
- > dbs
- ▼ hunderschule
 - > Casts
 - > Catalogs
 - > Event Triggers
 - > Extensions
 - > Foreign Data Wrappers
 - > Languages
 - > Publications
 - ▼ Schemas (1)
 - ▼ public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences

Query Editor Query History

```
1 --5- - Zu welchem Kurs sind durchschnittlich die meisten Doggen angemeldet?
2
3
4
5
6 --CREATE VIEW  anzahl_hunde(gkurs_id,kurs_id,Nr_hund) AS SELECT  gkurs_id ,kurs_id, count (hund_id) AS "Nr_hund"
7 --FROM  geplanteKurs GROUP BY gkurs_id ,kurs_id;
8
9 SELECT  k.kursname FROM anzahl_hunde a JOIN kurs k USING (kurs_id)
10 WHERE a.Nr_hund IN (SELECT MAX(Nr_hund) FROM anzahl_hunde);
```

Data Output Explain Messages Notifications

	kursname character varying (50)	
1	triks 2	

Data Export		Export 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



hunderschule/admin@dbs ▾

Query Editor Query History

```

1
2 --8 - - --name,rasse alle Hunde die in keine kurs teilnehmen und ihren Besitzern ;
3 SELECT besitzer.kn_id,besitzer.fullname,hunde.name,hunde.rasse FROM hunde INNER JOIN besitzer on hunde.kn_id=besitzer.kn_id
4 WHERE hund_id NOT IN (SELECT DISTINCT hund_id FROM geplantekurs)
5 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

- > Geschäftsreisen
- > Projektmanagment
- > Videothekenkette
- > dbs
- hunderschule
 - > Casts
 - > Catalogs
 - > Event Triggers
 - > Extensions
 - > Foreign Data Wrappers
 - > Languages
 - > Publications
 - public (1)
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures