

6/19/2021

## DATABASES

DATABASE SYSTEM FOR A PARCEL SHOP

Instructor: Prof. Dr. Ulrike Herster

<https://image.shutterstock.com>



Author: Emmanuel Duru

# TABLE OF CONTENTS

1.0	Introduction . . . . .	2
1.1	Entity Relation Diagram In Mc Notation . . . . .	.3
1.2	Relational Schema From The Entity Relation Diagram . . . . .	.4-10
1.3	Use cases with codes . . . . .	11-12
1.4	Aggregates . . . . .	12
1.5	Join . . . . .	.13-14
1.6	Grouping . . . . .	15
1.7	Views . . . . .	.16-17
1.8	Transactions . . . . .	.18
1.9	Trigger . . . . .	19

## 1.0                    **A DataBase system for my parcel store**

My parcel store database application holds a database system where i can manage all necessary information associated with my parcel store where a customer can pick up their parcels and send out parcels through my store which will be distributed to their destination. For every customer who is associated to my store has a preferred customer ID and every parcel is allocated to a registered customer ID for identification which is a principal access to their information in my database system.

Furthermore, my parcel store is situated in different areas in Hamburg and has a division in berlin and these branches are denoted with a certain identity, every store has an employee and holds records of every mobility event that holds for every corresponding incoming and out going parcels in the stores respectively.

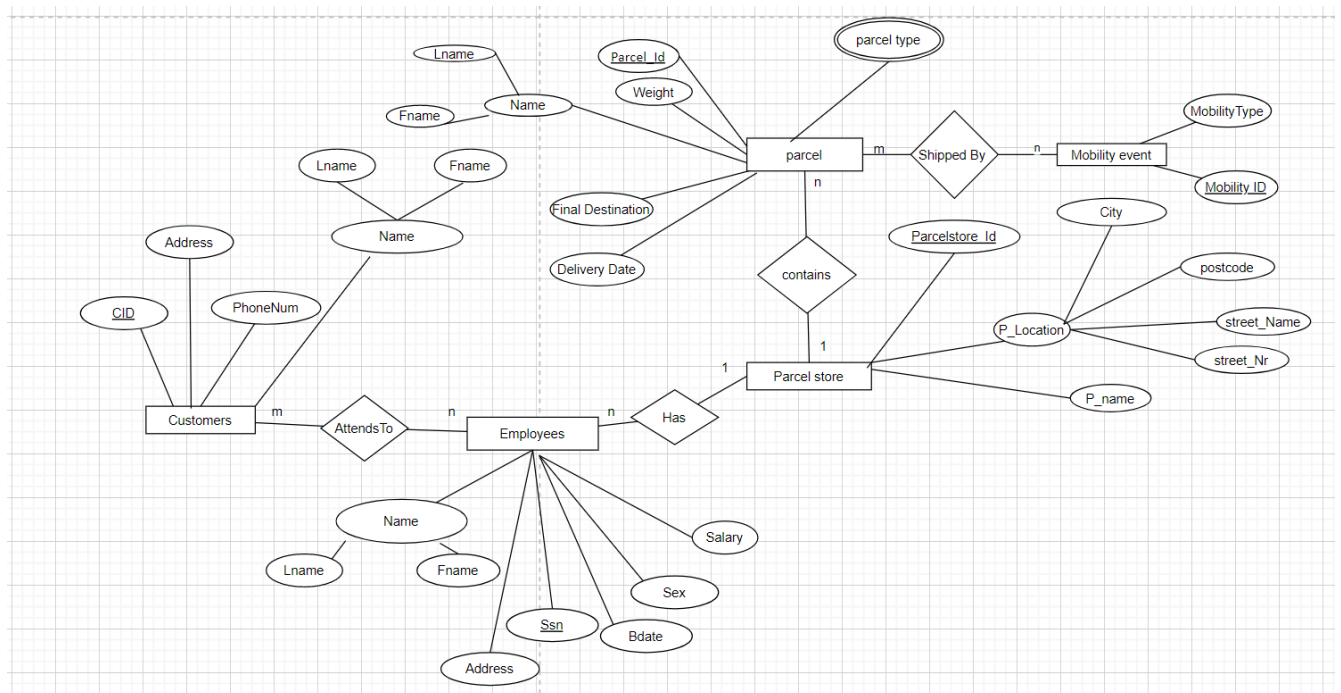
The use cases are stated accordingly in experiment 1.3 below with its corresponding code which was implemented in MYSQL.

### **Use Cases In The ANSEC Parcel Stores**

- As a parcel store owner i can delete or update customer details
- As a parcel store owner i can have a list of final destination of every parcel with its weights, and which store will it be delivered.
- As a parcel store owner i can have a list of all incoming parcels in the store either in descending or ascending orders.
- As a store owner i can have a list of my customer's name
- An employee salary can be updated.

## 1.1

## Entity Relation Diagram In Mc Notation



## 1.2 Relational Schema From The Entity Relation Diagram

parcels	<u>parcel_Id</u>	Weight	Final_destination	Delivery_date	<u>Parcelstore_Id(FK)</u>
---------	------------------	--------	-------------------	---------------	---------------------------

Mobility_event	<u>Mobility_Id</u>	Mobility_type		
----------------	--------------------	---------------	--	--

Parcel_Store	<u>Parcelstore_Id</u>	P_name	City	Postcode	Street_Name	Street_Nr
--------------	-----------------------	--------	------	----------	-------------	-----------

Employees	<u>Ssn</u>	Fname	Lname	E_Address	Bdate	Sex	Salary	<u>Parcelstore_Id(FK)</u>
-----------	------------	-------	-------	-----------	-------	-----	--------	---------------------------

Customers	<u>CID</u>	phoneNum	C_Address	Fname	Lname
-----------	------------	----------	-----------	-------	-------

Shipping	<u>Mobility_Id</u>	<u>lparcel_Id</u>
----------	--------------------	-------------------

Parcel type	<u>Parcel_Id(FK)</u>	<u>Parcel_type(PK)</u>	
-------------	----------------------	------------------------	--

create schema Parcel\_Store\_Project;

use parcel\_store\_project;

drop table Shipping;

drop table Employees;

drop table customer;

drop table parceltype;

drop table parcels;

drop table parcel\_store;

drop table Mobility\_event;

```

create table Parcel_Store
(
parcelstore_ID INT,

P_name VARCHAR(30)NOT NULL,

City VARCHAR(30) NOT NULL,

Postcode INT NOT NULL,

Street_name VARCHAR(30) NOT NULL,

Street_Nr INT NOT NULL,

primary key (parcelstore_ID)

);

```

	parcelstore_ID	P_name	City	Postcode	Street_name	Street_Nr
▶	204	ANSEC	Hamburg	20148	Bundestrasse	8
	205	ANSEC	Hamburg	20009	Alckermanstrasse	34
	206	ANSEC	Hamburg	20532	Altonastrasse	3
	207	ANSEC	Hamburg	22046	Eppendorfer Weg	21
	209	ANSEC	Hamburg	20428	gregstr	54
★	NULL	NULL	NULL	NULL	NULL	NULL

```

create table parcels
(
parcel_Id INT NOT NULL,

store_ID INT NOT NULL,

weight VARCHAR(10) NOT NULL,

final_destination VARCHAR(30) NOT NULL,

delivery_date DATE NOT NULL,

primary key(parcel_Id),

foreign key (store_ID) references Parcel_Store(parcelstore_ID)

);

```

	parcel_Id	store_ID	weight	final_destination	delivery_date
▶	1	204	8.6KG	8 Berliner Tor 23246 Hamburg	2021-04-13
	2	209	10.6KG	90 neindorfstr. 23246 Hamburg	2021-03-04
	3	205	5.6KG	12 frohmestr. 23246 Hamburg	2021-07-24
	4	206	8.6KG	21 sedanstr. 20146 Hamburg	2021-10-23
	5	207	63.6KG	21 heimfeldstr. 20146 Hamburg	2021-10-23
*	NULL	NULL	NULL	NULL	NULL

create table Mobility\_event

(

mobility\_ID INT NOT NULL,

mobility\_type VARCHAR(20),

primary key (mobility\_ID)

);

	mobility_ID	mobility_type
▶	102	A truck
	202	A car
	302	A bike
	402	A van
	502	A truck
*	NULL	NULL

create table Shipping

(

parcel\_ID INT NOT NULL,

mobility\_ID INT NOT NULL,

primary key (mobility\_ID,parcel\_ID),

foreign key (mobility\_ID) references Mobility\_event(mobility\_ID),

	parcel_ID	mobility_ID
▶	1	502
	2	402
	3	302
	4	202
	5	102
*	NULL	NULL

foreign key (parcel\_ID) references parcels(parcel\_ID)

);

create table Employees

(

store\_ID INT NOT NULL,

Essn INT NOT NULL,

Fname VARCHAR(20),

Lname VARCHAR(20),

E\_address VARCHAR(30),

E\_Bdate Date,

E\_Sex VARCHAR(10),

Salary INT NOT NULL,

primary key (Essn),

foreign key(store\_ID) references Parcel\_Store(parcelstore\_ID)

);

	store_ID	Essn	Fname	Lname	E_address	E_Bdate	E_Sex	Salary
▶	207	1765	lukaku	ogbonna	luognna@gmail.com	1990-04-05	male	23000
	209	2355	jude	ogbonna	jogbonna@gmail.com	1994-03-01	female	20000
	206	2425	margret	gerrad	stgerrad@gmail.com	1993-03-30	female	30000
	204	2615	jude	ogbonna	jogbonna@gmail.com	1994-03-01	male	20000
	205	3015	mathew	leonard	ml@gmail.com	1985-07-12	male	10000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

create table parceltype

(

parcel\_Id INT NOT NULL,

parcel\_type VARCHAR(15) NOT NULL,

primary key (parcel\_type,parcel\_Id),

foreign key (parcel\_id) references parcels(parcel\_Id)

);



	parcel_Id	parcel_type
▶	1	Incoming
	2	outgoing
	3	incoming
	4	Incoming
	5	Incoming
*	NULL	NULL

create table Customer

```
(
C_ID INT NOT NULL,
phoneNum VARCHAR(15)NOT NULL,
C_Address VARCHAR(30),
Fname VARCHAR(20),
Lname VARCHAR(20),
primary key (C_ID)
);
```

	C_ID	phoneNum	C_Address	Fname	Lname
▶	3	+491539197521	mittweg 23 20243 Hamburg	Austin	Ogu
	4	+491539197538	unnenstr. 9 20253 Hamburg	Hilary	Ogalagu
	6	+491599197438	quickbonstr. 9 20253 Hamburg	Renz	wolfgang
	8	+49149197038	lehmweg. 41 25332 Hamburg	Adriana	Duru
	10	+491539192338	unnenstr. 9 20253 Hamburg	Regina	Daniels
*	NULL	NULL	NULL	NULL	NULL

INSERT INTO customer VALUES

```
(004,'+491539197538','unnenstr. 9 20253 Hamburg','Hilary','Ogalagu'),
(010,'+491539192338','unnenstr. 9 20253 Hamburg','Regina','Daniels'),
(006,'+491599197438','quickbonstr. 9 20253 Hamburg','Renz','wolfgang'),
(008,'+49149197038','lehmweg. 41 25332 Hamburg','Adriana','Duru'),
(003,'+491539197521','mittweg 23 20243 Hamburg','Austin','Ogu');
```

INSERT INTO mobility\_event VALUES

```
(502,'A truck'),  
(402,'A van'),  
(302,'A bike'),  
(202,'A car'),  
(102,'A truck');
```

```
INSERT INTO Parcel_Store VALUES
```

```
(204,'ANSEC','Hamburg',20148,'Bundestrasse',8),  
(205,'ANSEC','Hamburg',20009,'Alckermanstrasse',34),  
(206,'ANSEC','Hamburg',20532,'Altonastrasse',3),  
(209,'ANSEC','Hamburg',20428,'gregstr',54),  
(207,'ANSEC','Hamburg',22046,'Eppendorfer Weg',21);
```

```
INSERT INTO employees VALUES
```

```
(204, 2615, 'jude','ogbonna','jogbonna@gmail.com',str_to_date('1994-03-01','%Y-%m-%d'),  
'male', '20000'),  
(209, 2355, 'jude','ogbonna','jogbonna@gmail.com',str_to_date('1994-03-01','%Y-%m-%d'),  
'female', '20000'),  
(205, 3015, 'mathew','leonard','ml@gmail.com',str_to_date('1985-07-12','%Y-%m-%d'),  
'male', '10000'),  
(206, 2425, 'margret','gerrad','stgerrad@gmail.com',str_to_date('1993-03-30','%Y-%m-%d'),  
'female', '30000'),  
(207, 1765, 'lukaku','ogbonna','luognna@gmail.com',str_to_date('1990-04-05','%Y-%m-%d'),  
'male', '23000');
```

```
INSERT INTO parcels VALUES
```

```
(001,204, '8.6KG', '8 Berliner Tor 23246 Hamburg',str_to_date('2021-04-13','%Y-%m-%d'));
```

INSERT INTO parcels VALUES

(002,209, '10.6KG', '90 neindorfstr. 23246 Hamburg',str\_to\_date('2021-03-04','%Y-%m-%d'));

INSERT INTO parcels VALUES

(003,205, '5.6KG', '12 frohmestr. 23246 Hamburg',str\_to\_date('2021-07-24','%Y-%m-%d'));

INSERT INTO parcels VALUES

(004,206, '8.6KG', '21 sedanstr. 20146 Hamburg',str\_to\_date('2021-10-23','%Y-%m-%d'));

INSERT INTO parcels VALUES

(005,207, '63.6KG', '21 heimfieldstr. 20146 Hamburg',str\_to\_date('2021-10-23','%Y-%m-%d'));

INSERT INTO parceltype VALUES

(001,'Incoming'),

(002,'outgoing'),

(004,'Incoming'),

(003,'incoming'),

(005,'Incoming');

INSERT INTO shipping VALUES

(001,502),

(002,402),

(003,302),

(004,202),

(005,102);

### 1.3: Use Cases In The ANSEC Parcel Stores with codes

- As a parcel store owner i can delete or update customer details

	C_ID	phoneNum	C_Address	Fname	Lname
▶	3	+491539197521	mittweg 23 20243 Hamburg	Austin	Ogu
	4	+491539197538	unnastr. 9 20253 Hamburg	Hilary	Ogalagu
	6	+491599197438	quickbonstr. 9 20253 Hamburg	Renz	wolfgang
	8	+49149197038	lehmweg. 41 25332 Hamburg	Adriana	Duru
*	NULL	NULL	NULL	NULL	NULL

	C_ID	phoneNum	C_Address	Fname	Lname
▶	3	+491539197521	mittweg 23 20243 Hamburg	Austin	Ogu
	4	+491539197538	unnastr. 9 20253 Hamburg	Hilary	Ogalagu
	6	+491599197438	quickbonstr. 9 20253 Hamburg	Renz	wolfgang
	8	+49149197038	lehmweg. 41 25332 Hamburg	Adriana	Duru
	10	+491539192338	unnastr. 9 20253 Hamburg	Regina	Daniels
*	NULL	NULL	NULL	NULL	NULL

delete from customer where C\_ID = 10;

- As a parcel store owner i can have a list of final destination of every parcel with its weights, and which store will it be delivered.

	store_ID	final_destination	weight
▶	204	8 Berliner Tor 23246 Hamburg	8.6KG
	209	90 neindorfstr. 23246 Hamburg	10.6KG
	205	12 frohmestr. 23246 Hamburg	5.6KG
	206	21 sedanstr. 20146 Hamburg	8.6KG
	207	21 heimfeldstr. 20146 Hamburg	63.6KG

select store\_ID,final\_destination,weight  
from parcels AS p  
join shipping AS s on p.parcel\_ID = s.parcel\_ID;

- As a store owner i can have a list of my customer name

	Fname	Lname
▶	Austin	Ogu
	Hilary	Ogalagu
	Renz	wolfgang
	Adriana	Duru

select Fname, Lname  
from customer;

- An employee salary can be updated.

	store_ID	Essn	Fname	Lname	E_address	E_Bdate	E_Sex	Salary
▶	207	1765	lukaku	ogbonna	luognna@gmail.com	1990-04-05	male	23000
	209	2355	jude	ogbonna	jogbonna@gmail.com	1994-03-01	female	35000
	206	2425	margret	gerrad	stgerrad@gmail.com	1993-03-30	female	30000
	204	2615	jude	ogbonna	jogbonna@gmail.com	1994-03-01	male	20000
	205	3015	matthew	leonard	ml@gmail.com	1985-07-12	male	10000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

UPDATE employees  
SET salary = 35000

Where Essn = 2355;

- I can have the list of all first name and last names of all my customers.

	Fname	Lname
▶	Austin	Ogu
	Hilary	Ogalagu
	Renz	wolfgang
	Adriana	Duru

```
select Fname, Lname
from customer;
```

- As a parcel store owner i can have a list of all incoming parcels in the store either in descending or ascending orders.

	parcel_Id	parcel_type
▶	5	Incoming
	4	Incoming
	3	incoming
	1	Incoming
*	NULL	NULL

	parcel_Id	parcel_type
▶	1	Incoming
	3	incoming
	4	Incoming
	5	Incoming
*	NULL	NULL

#### Ascending:

```
select* from parceltype where parcel_type = 'incoming'
order by parcel_ID asc;
```

#### Descending:

```
select* from parceltype where parcel_type = 'incoming'
order by parcel_ID desc;
```

## 1.4 Aggregates

#aggregates-----

```
select count(weight) from parcels;
```

```
select count(*)from employees where salary>10000;
```

```
SELECT min(salary) AS salary FROM employees;
```

```
SELECT max(salary) AS salary FROM employees;
```

## 1.5

## Join

#-----to achieve the parcel final destination of every incoming and outgoing parcels with the mobility event and if its an incoming or outgoing parcels-----

select

parcels.store\_ID,parcel\_store.P\_name,parcel\_store.City,shipping.mobility\_ID,  
mobility\_event.mobility\_type, parceltype.parcel\_type,  
parcels.delivery\_date,parcels.final\_destination

from parcels,parcel\_store,mobility\_event, parceltype

inner join shipping

where parcels.parcel\_Id = shipping.parcel\_Id and parcels.store\_ID =  
parcel\_store.parcelstore\_ID and mobility\_event.mobility\_ID =  
shipping.mobility\_ID and parceltype.parcel\_Id = parcels.parcel\_Id;

	store_ID	P_name	City	mobility_ID	mobility_type	parcel_type	delivery_date	final_destination
►	204	ANSEC	Hamburg	502	A truck	Incoming	2021-04-13	8 Berliner Tor 23246 Hamburg
	209	ANSEC	Hamburg	402	A van	outgoing	2021-03-04	90 neindorfstr. 23246 Hamburg
	205	ANSEC	Hamburg	302	A bike	incoming	2021-07-24	12 frohmestr. 23246 Hamburg
	206	ANSEC	Hamburg	202	A car	Incoming	2021-10-23	21 sedanstr. 20146 Hamburg
	207	ANSEC	Hamburg	102	A truck	Incoming	2021-10-23	21 heimfeldstr. 20146 Hamburg

#---gets the list of all customers information, their parcel\_id's and if its an incoming parcel or outgoing parcel-----

select\* from customer

left join parceltype

on customer.C\_ID = parceltype.parcel\_Id;

	C_ID	phoneNum	C_Address	Fname	Lname	parcel_Id	parcel_type
►	1	+491560097538	magretstr. 40 34353 berlin	dele	abel	1	Incoming
	2	+49156009638	allinstr. 40 2030 berlin	zoin	nen	2	outgoing
	3	+491539197521	mittweg 23 20243 Hamburg	Austin	Ogu	3	incoming
	4	+491539197538	unnastr. 9 20253 Hamburg	Hilary	Ogalagu	4	Incoming
	5	+491530097538	billstr. 9 20253 Hamburg	kola	steph	5	Incoming
	6	+491599197438	quickbonstr. 9 20253 Hamburg	Renz	wolfgang	NULL	NULL
	8	+49149197038	lehmweg. 41 25332 Hamburg	Adriana	Duru	NULL	NULL
	10	+491539192338	unnastr. 9 20253 Hamburg	Regina	Daniels	NULL	NULL

##-----to get the delivery date and which parcel was delivered to a customer-----

```
select delivery_date, Fname,Lname,C_address, parcel_Id
from customer join parcels
on C_ID = parcel_Id
order by Fname;
```

	delivery_date	Fname	Lname	C_address	parcel_Id
►	2021-07-24	Austin	Ogu	mittweg 23 20243 Hamburg	3
	2021-04-13	dele	abel	magretstr. 40 34353 berlin	1
	2021-10-23	Hilary	Ogalagu	unnastr. 9 20253 Hamburg	4
	2021-10-23	kola	steph	billstr. 9 20253 Hamburg	5
	2021-03-04	zoin	nena	allinstr. 40 2030 berlin	2

#join-----gets list of which customer has a parcel in a store considering if its an outgoing or incoming parcel-----

```
select customer.C_ID,customer.Fname,customer.Lname,customer.C_address,
parcetype.parcel_type, parcels.store_ID, parcels.parcel_Id
```

From customer, parcetype

inner join parcels

```
where customer.C_ID = parcetype.parcel_Id and parcetype.parcel_Id =
parcels.parcel_Id;
```

	C_ID	Fname	Lname	C_address	parcel_type	store_ID	parcel_Id
►	1	dele	abel	magretstr. 40 34353 berlin	Incoming	204	1
	3	Austin	Ogu	mittweg 23 20243 Hamburg	incoming	205	3
	4	Hilary	Ogalagu	unnastr. 9 20253 Hamburg	Incoming	206	4
	5	kola	steph	billstr. 9 20253 Hamburg	Incoming	207	5
	2	zoin	nena	allinstr. 40 2030 berlin	outgoing	209	2

## 1.6                      **Grouping**

#Grouping-----

```
select delivery_date,sum(weight) AS total_day_weight
```

```
From parcels
```

```
group by delivery_date;
```

	delivery_date	total_day_weight
▶	2021-04-13	8.6
	2021-03-04	10.6
	2021-07-24	5.6
	2021-10-23	72.2



## 1.7 Special cases with views

#Views-----

A view that represents incoming and outgoing parcels in the store.

```
drop view delivery;
create view Delivery
AS select p.store_ID, p.final_destination , pt.parcel_type, p.delivery_date
from parcels p, parceltype pt
where p.parcel_Id = pt.parcel_Id;
```

	store_ID	final_destination	parcel_type	delivery_date
►	204	8 Berliner Tor 23246 Hamburg	Incoming	2021-04-13
	209	90 neindorfstr. 23246 Hamburg	outgoing	2021-03-04
	205	12 frohmestr. 23246 Hamburg	incoming	2021-07-24
	206	21 sedanstr. 20146 Hamburg	Incoming	2021-10-23
	207	21 heimfeldstr. 20146 Hamburg	Incoming	2021-10-23

#-----

create view parcelstore\_addresses

AS select ps.Street\_name, ps.Street\_Nr, ps.city

from parcel\_store ps;

	Street_name	Street_Nr	city
►	Bundestrasse	8	Hamburg
	Alckermanstrasse	34	Hamburg
	Altonastrasse	3	Hamburg
	Eppendorfer Weg	21	Hamburg
	alexanderstr	10	Berlin
	gregstr	54	Hamburg
	solastr	4	Hamburg

```
create view parcelstore_addresses
AS select ps.Street_name, ps.Street_Nr, ps.city
from parcel_store ps; |
```

#-----

create view high\_salary as

select count(\*)from employees where salary>10000;

```
create view high_salary as  
select count(*) from employees where salary > 10000;
```

	count(*)
▶	5

## 1.8 Transactions

#Transaction-----

```
START TRANSACTION;
```

```
-- 2. Get the highest salary
```

```
SELECT @salary:= MAX(salary) FROM employees;
```

```
INSERT INTO employees(store_ID, Essn, Fname, Lname,E_address,E_Bdate,E_Sex, salary)
```

```
VALUES (301, 4334, 'Frank','lampard', 'frank@gmail.com', str_to_date('1994-03-01','%Y-%m-%d'), 'male', '90000');
```

```
-- 4. I Inserted a new record into the parcel_store table
```

```
INSERT INTO parcel_store(parcelstore_ID, P_name, City, Postcode,Street_name,Street_Nr)
```

```
VALUES (208, 'ANSEC', 'Berlin',38219,'alexanderstr', 10);
```

```
-- 5. then i Committed the changes
```

```
COMMIT;
```

	parcelstore_ID	P_name	City	Postcode	Street_name	Street_Nr
▶	204	ANSEC	Hamburg	20148	Bundestrasse	8
	205	ANSEC	Hamburg	20009	Aldkermanstrasse	34
	206	ANSEC	Hamburg	20532	Altonastrasse	3
	207	ANSEC	Hamburg	22046	Eppendorfer Weg	21
	208	ANSEC	Berlin	38219	alexanderstr	10
	209	ANSEC	Hamburg	20428	gregstr	54
*	NULL	NULL	NULL	NULL	NULL	NULL

	store_ID	Essn	Fname	Lname	E_address	E_Bdate	E_Sex	Salary
▶	207	1765	lukaku	ogbonna	luognna@gmail.com	1990-04-05	male	23000
	209	2355	jude	ogbonna	jogbonna@gmail.com	1994-03-01	female	35000
	206	2425	margret	gerrad	stgerrad@gmail.com	1993-03-30	female	30000
	204	2615	jude	ogbonna	jogbonna@gmail.com	1994-03-01	male	20000
	205	3015	mathew	leonard	ml@gmail.com	1985-07-12	male	10000
	206	4334	Frank	lampard	frank@gmail.com	1994-03-01	male	90000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table representing the committed values of the transaction (parcel\_store and employee)

## 1.9 Triggers

#Trigger-----

```
DROP TRIGGER IF EXISTS emp_salary;
```

```
DELIMITER //
```

```
Create Trigger before_insert_emp_salary
```

```
BEFORE INSERT ON employees FOR EACH ROW
```

```
BEGIN
```

```
IF NEW.salary < 5000 THEN SET NEW.salary = 0;
```

```
END IF;
```

```
END //
```

```
INSERT INTO employees VALUES
```

```
(207, 6352, 'Nena', 'Hensel', '14 dammtorstr 22145 hamburg', str_to_date('1990-04-05', '%Y-%m-%d'), 'female', '3000');
```

store_ID	Essn	Fname	Lname	E_address	E_Bdate	E_Sex	Salary
207	1765	lukaku	ogbonna	luogonna@gmail.com	1990-04-05	male	23000
210	2252	Drake	Hensel	drakehe@gmail.com	1990-05-05	male	0
209	2355	jude	ogbonna	jogbonna@gmail.com	1994-03-01	female	35000
206	2425	margret	gerrad	stgerrad@gmail.com	1993-03-30	female	30000
204	2615	jude	ogbonna	jogbonna@gmail.com	1994-03-01	male	20000
205	3015	mathew	leonard	nl@gmail.com	1985-07-12	male	10000
206	4334	Frank	lampard	frank@gmail.com	1994-03-01	male	90000
207	6352	Nena	Hensel	14 dammtorstr 22145 hamburg	1990-04-05	female	0

**Note:** the above code and table shows that trigger works, thus for every employee salary which is less than 5000 it will display zero in the row automatically.

## REFERENCES

<https://image.shutterstock.com/image-vector/big-data-visualization-abstract-database-260nw-796583365.jpg>