Exercise 1

- Te table is in **1NF**, so I will make **3NF**

```
-- Creation of the tables

CREATE TABLE Customer
(
  customerId INT PRIMARY KEY,
  customerName VARCHAR(30),
  city VARCHAR(30)
);

CREATE TABLE Item
(
  itemId INT PRIMARY KEY,
  itemName VARCHAR(30),
  price DECIMAL(7,2)
);

CREATE TABLE "Order"
(
  orderId INT PRIMARY KEY,
  "date" VARCHAR(30)
);
CREATE TABLE common
(
  ord INT NOT NULL PRIMARY KEY,
  oid INT,
  quant INT,
  iId INT,
  cId INT,
  FOREIGN KEY (oId) REFERENCES "Order"(orderid),
  FOREIGN KEY (iId) REFERENCES item(itemid),
  FOREIGN KEY (cId) REFERENCES customer(customerid)
);

-- Fillig the tables with data

INSERT INTO customer (customerid, customername, city)
VALUES (101, 'Martin', 'Prague'),
(107, 'Herman', 'Madrid'),
(110, 'Pedro', 'Moscow');

INSERT INTO item (itemid, itemname, price)
VALUES (3786, 'Net', 35.00),
(4011, 'Racket', 65.00),
(9132, 'Pack-3', 4.75),
(5794, 'Pack-6', 5.00),
(3141, 'Cover', 10.00);

INSERT INTO "Order" (orderid, "date", cid)
VALUES (2301, '23/02/2011', 101),
(2302, '25/02/2011', 107),
(2303, '27/02/2011', 110);

INSERT INTO common (ord, oid, iid, quant)
VALUES (1, 2301, 3786, 3),
(2, 2301, 4011, 6),
(3, 2301, 9132, 8),
(4, 2302, 5794, 4),
(5, 2303, 4011, 2),
(6, 2303, 3141, 2);
```

Table **customer**:

customerid	customername	city
101	Martin	Prague
107	Herman	Madrid
110	Pedro	Moscow

Table **item**:

itemid	itemname	price
3786	Net	35.00
4011	Racket	65.00
9132	Pack-3	4.75
5794	Pack-6	5.00
3141	Cover	10.00

Table **Order**:

orderid	date	cid
2301	23/02/2011	101
2302	25/02/2011	107
2303	27/02/2011	110

Table **common**:

ord	oid	quant	iid
1	2301	3	3786
2	2301	6	4011
3	2301	8	9132
4	2302	4	5794
5	2303	2	4011
6	2303	2	3141

Queries for the result table 1

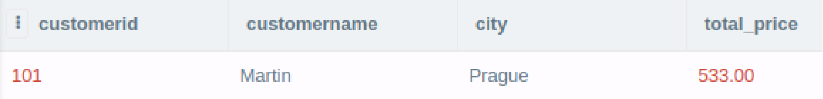
- **Calculate the total number of items per order and the total amount to pay for the order**

```
SELECT O.oid, SUM(O.quant) as total_quant, SUM(total) as total_price
FROM (
  SELECT oid, quant, price, SUM(quant*price) as Total
  FROM (common INNER JOIN item ON common.iid = item.itemid) L
  Group BY oid, quant, price
  ) O
GROUP BY O.oid
```

oid	total_quant	total_price
2302	4	20.00
2303	4	150.00
2301	17	533.00

- **Obtain the customer whose purchase in terms of money has been greater than the others**

```
SELECT customerid, customername, city, total_price
FROM (
  SELECT Al3.oid, Al3.cid, SUM(total) as total_price
  FROM ((
    SELECT oid, quant, price, SUM(quant*price) as Total
    FROM (common INNER JOIN item ON common.iid = item.itemid) L
    Group BY oid, quant, price
    ) Al INNER JOIN "Order" ON Al.oid = "Order".orderid) Al3
  GROUP BY Al3.oid, Al3.cid
  HAVING SUM(total) = (
    SELECT MAX(total_price) as maximum
    FROM (
      SELECT O.oid, SUM(O.quant) as total_quant, SUM(total) as total_price
      FROM (
        SELECT oid, quant, price, SUM(quant*price) as Total
        FROM (common INNER JOIN item ON common.iid = item.itemid) L
        Group BY oid, quant, price
        ) O
      GROUP BY O.oid
    ) Alll
  )
) cus INNER JOIN customer ON cus.cid = customer.customerid
```



customerid	customername	city	total_price
101	Martin	Prague	533.00

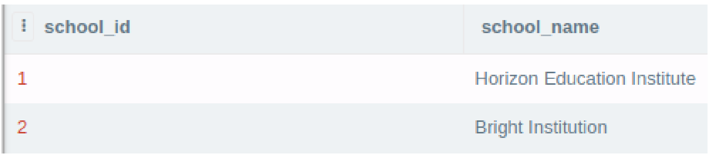
Exercise 2

- Te table is in **1NF**, so I will make **3NF**

```
-- Creation of the tables
CREATE TABLE school
(
  school_id SERIAL Primary Key,
  school_name VARCHAR(50)
);

CREATE TABLE room
(
  room_id SERIAL PRIMARY KEY,
  room_name Varchar(10)
);

CREATE TABLE teacher
(
  teacher_id SERIAL PRIMARY KEY,
  teacher_name VARCHAR(50)
);

CREATE TABLE course
(
  course_id SERIAL PRIMARY KEY,
  course_name VARCHAR(50)
);

CREATE TABLE publisher
(
  publisher_id SERIAL PRIMARY KEY,
  publisher_name VARCHAR(30)
);

create table grade
(
  grade_id SERIAL PRIMARY KEY,
  grade_name VARCHAR(20)
```

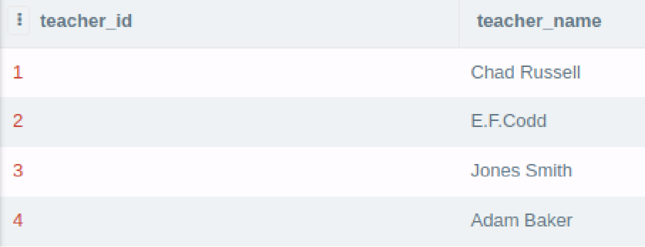
```sql
);

CREATE TABLE book
(
  book_id SERIAL PRIMARY KEY,
  book_name VARCHAR(50),
  p_id INT,
  FOREIGN KEY (p_id) REFERENCES publisher(publisher_id)
);

CREATE TABLE common
(
  common_id SERIAL PRIMARY KEY,
  s_id INT,
  t_id INT,
  c_id INT,
  r_id INT,
  g_id INT,
  b_id INT,
  loan_date VARCHAR(20),
  FOREIGN KEY (s_id) REFERENCES school(school_id),
  FOREIGN KEY (t_id) REFERENCES teacher(teacher_id),
  FOREIGN KEY (c_id) REFERENCES course(course_id),
  FOREIGN KEY (r_id) REFERENCES room(room_id),
  FOREIGN KEY (g_id) REFERENCES grade(grade_id),
  FOREIGN KEY (b_id) REFERENCES book(book_id)
);

-- Filling the tables with data

INSERT INTO school (school_name) VALUES
('Horizon Education Institute'),
('Bright Institution');
INSERT INTO teacher (teacher_name) VALUES
('Chad Russell'),
('E.F.Codd'),
('Jones Smith'),
('Adam Baker');
INSERT INTO publisher (publisher_name) VALUES
('BOA Editions'),
('Taylor & Francis Publishing'),
('Prentice Hall'),
('McGraw Hill');
INSERT INTO room (room_name) VALUES
('1.A01'),
('1.B01'),
('2.B01');
INSERT INTO grade (grade_name) VALUES
('1st grade'),
('2nd grade');
INSERT iNTO book (book_name, p_id) VALUES
('Learning and teaching in early childhood', 1),
('Preschool,N56', 2),
('Early Childhood Education N9', 3),
('Know how to educate: guide for Parents', 4);
INSERT INTO course (course_name) VALUES
('Logical thinking'),
('Wrtting'),
('Numerical Thinking'),
('Spatial, Temporal and Causal Thinking'),
('English');
INSERT INTO common (s_id, t_id, c_id, r_id, g_id, b_id, loan_date) VALUES
(1, 1, 1, 1, 1, 1, '09/09/2010'),
(1, 1, 2, 1, 1, 2, '05/05/2010'),
(1, 1, 3, 1, 1, 1, '05/05/2010'),
(1, 2, 4, 2, 1, 3, '06/05/2010'),
(1, 2, 3, 2, 1, 1, '06/05/2010'),
(1, 3, 2, 1, 2, 1, '09/09/2010'),
(1, 3, 5, 1, 2, 4, '05/05/2010'),
(2, 4, 1, 3, 1, 4, '18/12/2010'),
(2, 4, 3, 3, 1, 1, '06/05/2010');
```

Table **school**:



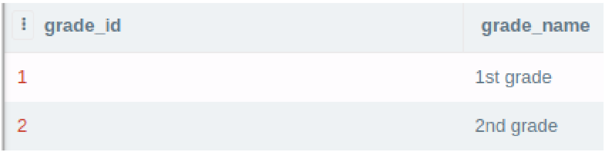
school_id	school_name
1	Horizon Education Institute
2	Bright Institution

Table **teacher**:



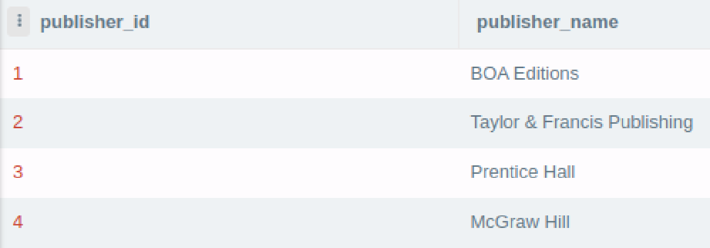
teacher_id	teacher_name
1	Chad Russell
2	E.F.Codd
3	Jones Smith
4	Adam Baker

Table **grade**:



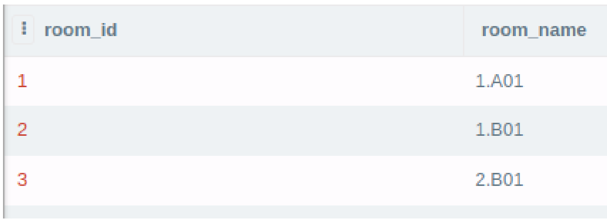
grade_id	grade_name
1	1st grade
2	2nd grade

Table **publisher**:



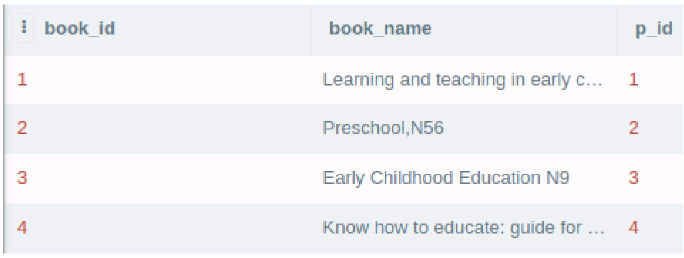
publisher_id	publisher_name
1	BOA Editions
2	Taylor & Francis Publishing
3	Prentice Hall
4	McGraw Hill

Table **room**:



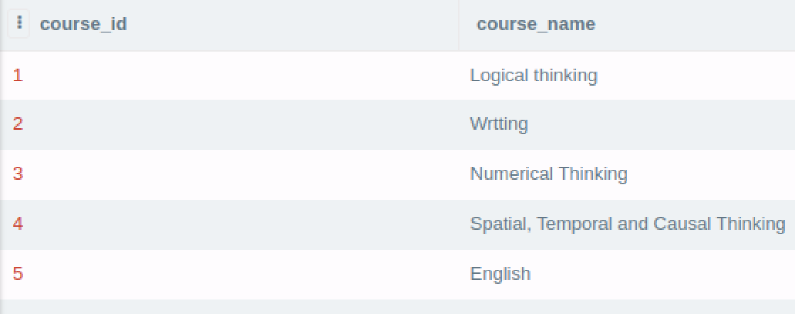
room_id	room_name
1	1.A01
2	1.B01
3	2.B01

Table **book**:



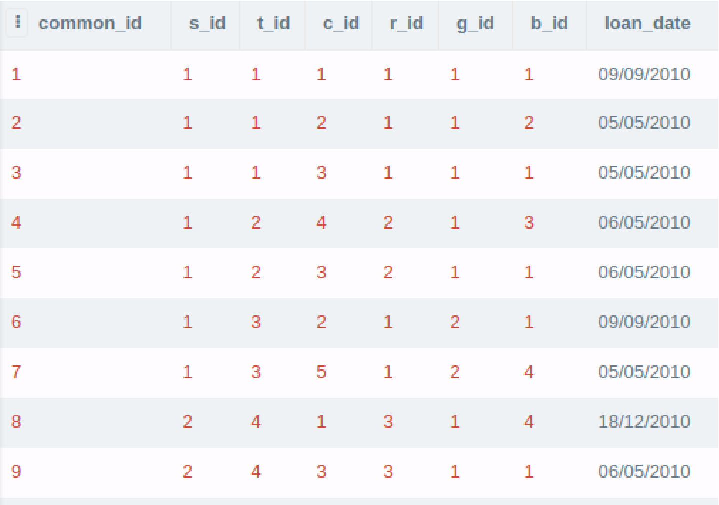
book_id	book_name	p_id
1	Learning and teaching in early c…	1
2	Preschool,N56	2
3	Early Childhood Education N9	3
4	Know how to educate: guide for …	4

Table **course**:



course_id	course_name
1	Logical thinking
2	Wrtting
3	Numerical Thinking
4	Spatial, Temporal and Causal Thinking
5	English

Table **common**:

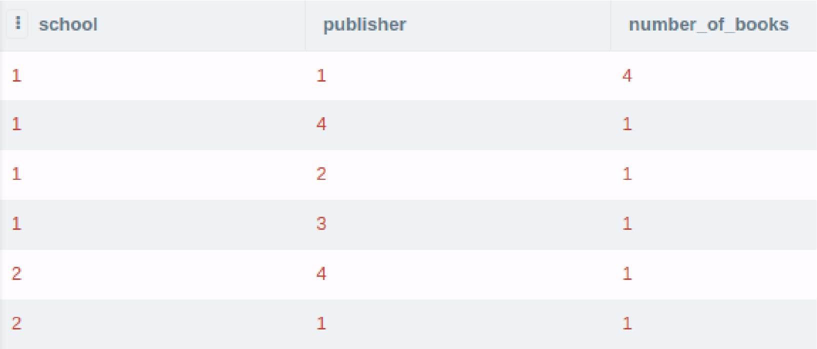


common_id	s_id	t_id	c_id	r_id	g_id	b_id	loan_date
1	1	1	1	1	1	1	09/09/2010
2	1	1	2	1	1	2	05/05/2010
3	1	1	3	1	1	1	05/05/2010
4	1	2	4	2	1	3	06/05/2010
5	1	2	3	2	1	1	06/05/2010
6	1	3	2	1	2	1	09/09/2010
7	1	3	5	1	2	4	05/05/2010
8	2	4	1	3	1	4	18/12/2010
9	2	4	3	3	1	1	06/05/2010

Queries for the result table 1

- **Obtain for each of the schools, the number of books that have been loaned to each publishers**

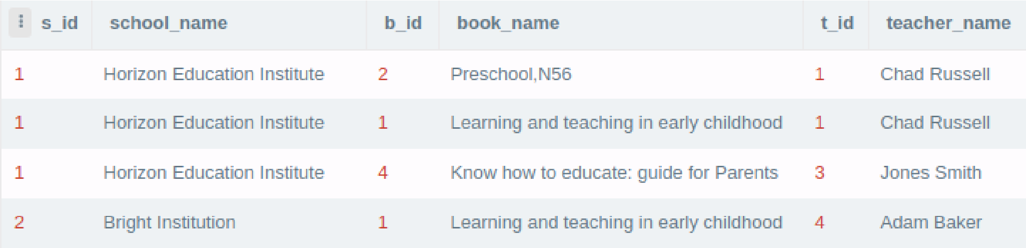
```
SELECT s_id as school, p_id as publisher, COUNT(O.loan_date) as number_of_books
FROM (common INNER JOIN book ON common.b_id = book.book_id) O
GROUP BY O.s_id, O.p_id
```



school	publisher	number_of_books
1	1	4
1	4	1
1	2	1
1	3	1
2	4	1
2	1	1

- **For each school, find the book that has been on loan the longest and the teacher in charge of it**

```
SELECT V5.s_id, V5.school_name, V5.b_id, V5.book_name, V5.t_id, V5.teacher_name
FROM (((((
  SELECT s_id as sch, MIN(loan_date) as ld
  FROM common
  GROUP BY common.s_id
) V1 INNER JOIN common ON (common.s_id = V1.sch AND common.loan_date = V1.ld)) V2
INNER JOIN teacher ON V2.t_id = teacher.teacher_id) V3
INNER JOIN book ON V3.b_id = book.book_id) V4
INNER JOIN school ON V4.s_id = school.school_id) V5
```



s_id	school_name	b_id	book_name	t_id	teacher_name
1	Horizon Education Institute	2	Preschool,N56	1	Chad Russell
1	Horizon Education Institute	1	Learning and teaching in early childhood	1	Chad Russell
1	Horizon Education Institute	4	Know how to educate: guide for Parents	3	Jones Smith
2	Bright Institution	1	Learning and teaching in early childhood	4	Adam Baker