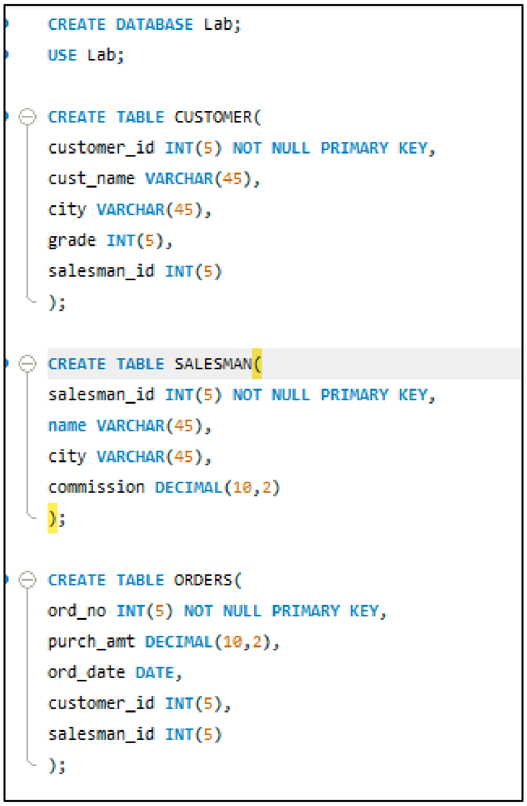
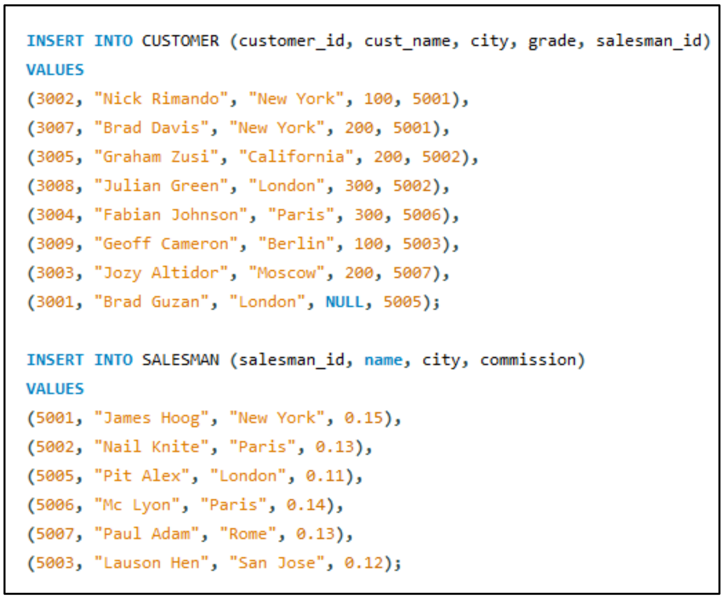
Henreich Catig
BSIT – 3B

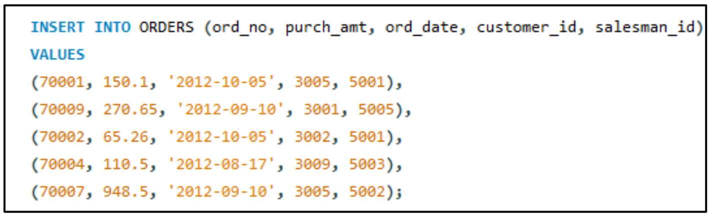
1. Create the following tables



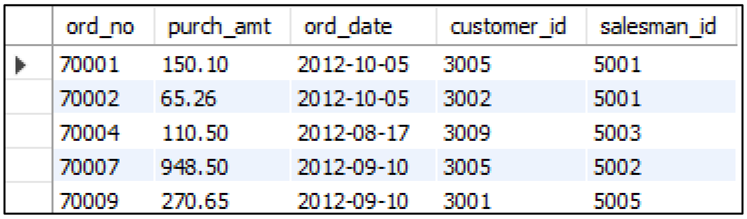
```
CREATE DATABASE Lab;
USE Lab;

CREATE TABLE CUSTOMER(
customer_id INT(5) NOT NULL PRIMARY KEY,
cust_name VARCHAR(45),
city VARCHAR(45),
grade INT(5),
salesman_id INT(5)
);

CREATE TABLE SALESMAN(
salesman_id INT(5) NOT NULL PRIMARY KEY,
name VARCHAR(45),
city VARCHAR(45),
commission DECIMAL(10,2)
);

CREATE TABLE ORDERS(
ord_no INT(5) NOT NULL PRIMARY KEY,
purch_amt DECIMAL(10,2),
ord_date DATE,
customer_id INT(5),
salesman_id INT(5)
);
```



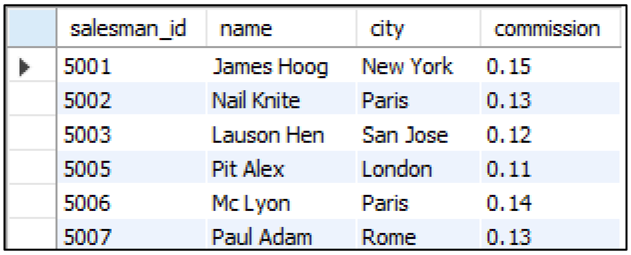
```
INSERT INTO CUSTOMER (customer_id, cust_name, city, grade, salesman_id)
VALUES
(3002, "Nick Rimando", "New York", 100, 5001),
(3007, "Brad Davis", "New York", 200, 5001),
(3005, "Graham Zusi", "California", 200, 5002),
(3008, "Julian Green", "London", 300, 5002),
(3004, "Fabian Johnson", "Paris", 300, 5006),
(3009, "Geoff Cameron", "Berlin", 100, 5003),
(3003, "Jozy Altidor", "Moscow", 200, 5007),
(3001, "Brad Guzan", "London", NULL, 5005);

INSERT INTO SALESMAN (salesman_id, name, city, commission)
VALUES
(5001, "James Hoog", "New York", 0.15),
(5002, "Nail Knite", "Paris", 0.13),
(5005, "Pit Alex", "London", 0.11),
(5006, "Mc Lyon", "Paris", 0.14),
(5007, "Paul Adam", "Rome", 0.13),
(5003, "Lauson Hen", "San Jose", 0.12);
```



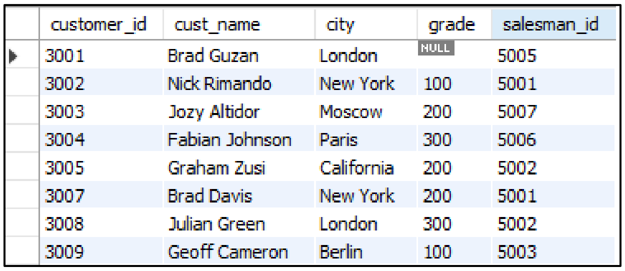
```
INSERT INTO ORDERS (ord_no, purch_amt, ord_date, customer_id, salesman_id)
VALUES
(70001, 150.1, '2012-10-05', 3005, 5001),
(70009, 270.65, '2012-09-10', 3001, 5005),
(70002, 65.26, '2012-10-05', 3002, 5001),
(70004, 110.5, '2012-08-17', 3009, 5003),
(70007, 948.5, '2012-09-10', 3005, 5002);
```



ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.10	2012-10-05	3005	5001
70002	65.26	2012-10-05	3002	5001
70004	110.50	2012-08-17	3009	5003
70007	948.50	2012-09-10	3005	5002
70009	270.65	2012-09-10	3001	5005

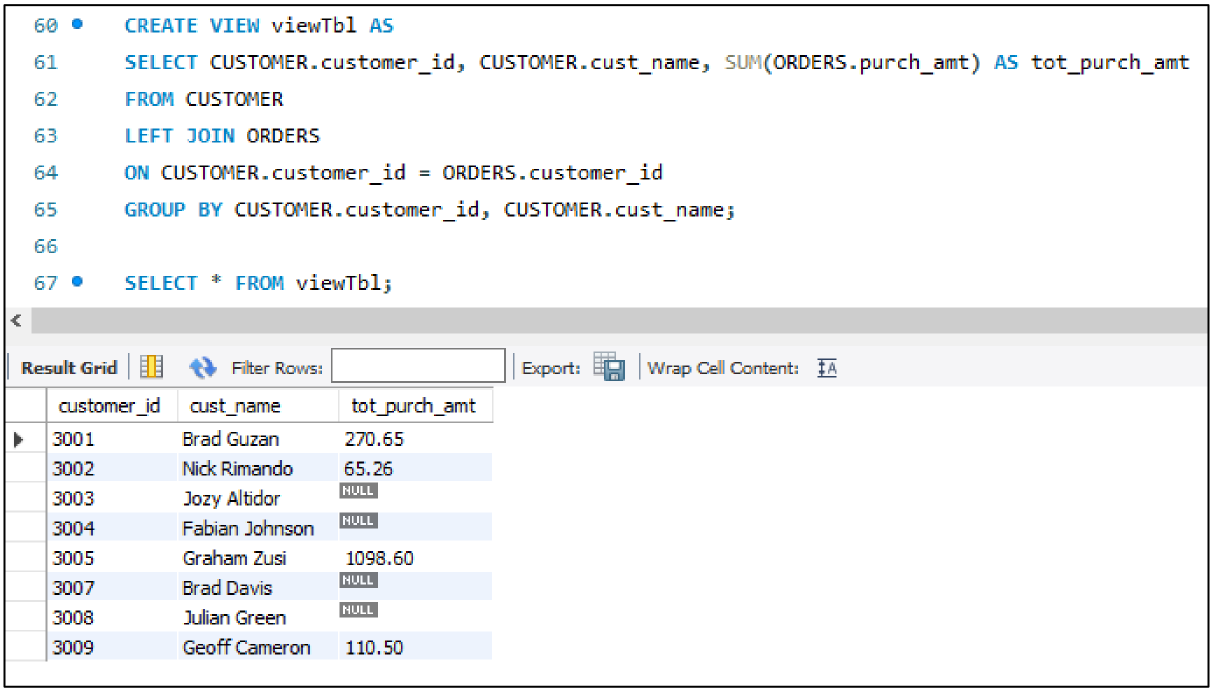


salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5003	Lauson Hen	San Jose	0.12
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13



customer_id	cust_name	city	grade	salesman_id
3001	Brad Guzan	London	NULL	5005
3002	Nick Rimando	New York	100	5001
3003	Jozy Altidor	Moscow	200	5007
3004	Fabian Johnson	Paris	300	5006
3005	Graham Zusi	California	200	5002
3007	Brad Davis	New York	200	5001
3008	Julian Green	London	300	5002
3009	Geoff Cameron	Berlin	100	5003

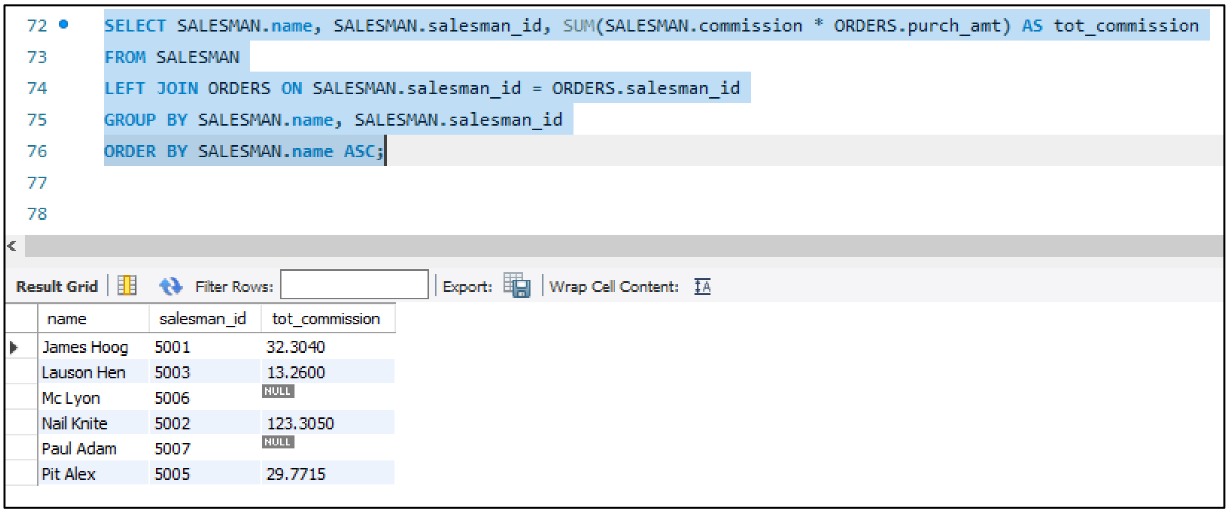
2. Create a **View** containing the total purchased amount of every customer. Show **Customer ID, customer name, and purchase amount for every amount**



```
CREATE VIEW viewTbl AS
SELECT CUSTOMER.customer_id, CUSTOMER.cust_name, SUM(ORDERS.purch_amt) AS tot_purch_amt
FROM CUSTOMER
LEFT JOIN ORDERS
ON CUSTOMER.customer_id = ORDERS.customer_id
GROUP BY CUSTOMER.customer_id, CUSTOMER.cust_name;

SELECT * FROM viewTbl;
```

customer_id	cust_name	tot_purch_amt
3001	Brad Guzan	270.65
3002	Nick Rimando	65.26
3003	Jozy Altidor	NULL
3004	Fabian Johnson	NULL
3005	Graham Zusi	1098.60
3007	Brad Davis	NULL
3008	Julian Green	NULL
3009	Geoff Cameron	110.50

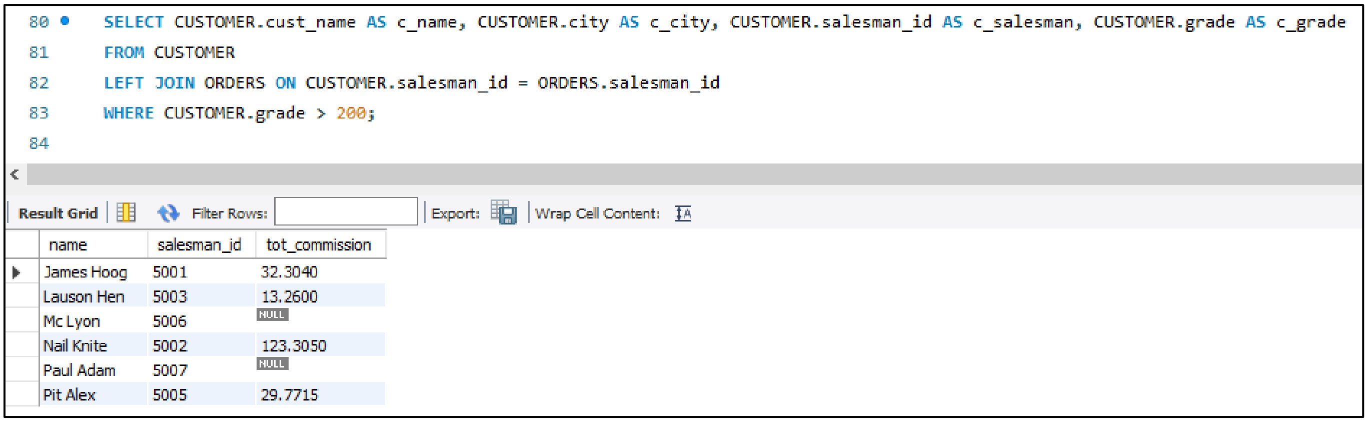
3. Compute for the **total commission** of each salesman. The result should be **sorted** by ascending order of the Salesman's name. Total Commission is equal to commission * total purchased amount. Return **Salesman Name, Salesman ID, and Total Commission**



```
SELECT SALESMAN.name, SALESMAN.salesman_id, SUM(SALESMAN.commission * ORDERS.purch_amt) AS tot_commission
FROM SALESMAN
LEFT JOIN ORDERS ON SALESMAN.salesman_id = ORDERS.salesman_id
GROUP BY SALESMAN.name, SALESMAN.salesman_id
ORDER BY SALESMAN.name ASC;
```

name	salesman_id	tot_commission
James Hoog	5001	32.3040
Lauson Hen	5003	13.2600
Mc Lyon	5006	NULL
Nail Knite	5002	123.3050
Paul Adam	5007	NULL
Pit Alex	5005	29.7715

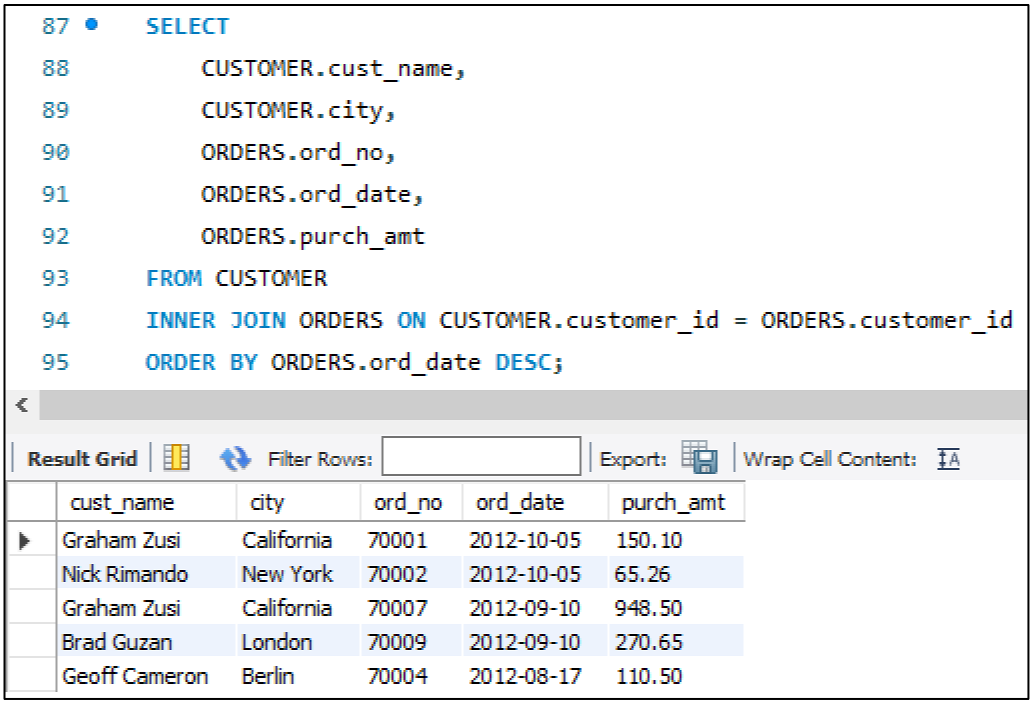
4. List down the **salesperson(s) and the customer(s)** each of them represents with **grade above 200**. Return Custom Name, city, Salesman, and grade



```sql
SELECT CUSTOMER.cust_name AS c_name, CUSTOMER.city AS c_city, CUSTOMER.salesman_id AS c_salesman, CUSTOMER.grade AS c_grade
FROM CUSTOMER
LEFT JOIN ORDERS ON CUSTOMER.salesman_id = ORDERS.salesman_id
WHERE CUSTOMER.grade > 200;
```

name	salesman_id	tot_commission
James Hoog	5001	32.3040
Lauson Hen	5003	13.2600
Mc Lyon	5006	NULL
Nail Knite	5002	123.3050
Paul Adam	5007	NULL
Pit Alex	5005	29.7715

5. Determine all the orders made by existing customers by making a report with the **customer name, city on number, order date, and order amount**. **Sort** all orders of each customer according to the order date in descending order.



```sql
SELECT
    CUSTOMER.cust_name,
    CUSTOMER.city,
    ORDERS.ord_no,
    ORDERS.ord_date,
    ORDERS.purch_amt
FROM CUSTOMER
INNER JOIN ORDERS ON CUSTOMER.customer_id = ORDERS.customer_id
ORDER BY ORDERS.ord_date DESC;
```

cust_name	city	ord_no	ord_date	purch_amt
Graham Zusi	California	70001	2012-10-05	150.10
Nick Rimando	New York	70002	2012-10-05	65.26
Graham Zusi	California	70007	2012-09-10	948.50
Brad Guzan	London	70009	2012-09-10	270.65
Geoff Cameron	Berlin	70004	2012-08-17	110.50