



Practice Assignment 5

Instruction:

** Students are allowed to write their answers (like SQL queries, Screen shot of outputs, etc.) in word file (Answer sheet) provided by instructor. After finishing the assignment, students must convert the word file (Answer sheet) into a PDF file. Finally, students upload the file in Moodle.*

1. Create the following tables in a new database 'Assignment3':

Clients(**Client Number**, Client_Name, Address, City, Pincode, Province, Amount_Paid, Amount_Due)

Product(**Product Number**, Product_Name, Quantity_On_Hand, Quantity_Sell, Sell_Price, Cost_Price)

Salesman (**Salesman Number**, Salesman_Name, Address, City, Pincode, Province, Salary, Sales_Target, Target_Achieve, Phone)

Salesorder(**Order Number**, Order_Date, **Client Number**, **Salesman Number**, Delivery_Status, Delivery_Date, Order_Status)

Salesorderdetails(**Order Number**, **Product Number**, Order_Quantity)

Insert more values data below:

Salesman

('S007','Quang','Chanh My','Da Lat',700032,'Lam Dong',25000,90,95,'0900853487')

('S008','Hoa','Hoa Phu','Thu Dau Mot',700051,'Binh Duong',13500,50,75,'0998213659')

Salesorder

('O20015','2022-05-12','C108','S007','On Way', '2022-05-15','Successful')

('O20016','2022-05-16','C109','S008','Ready to Ship',null,'In Process')

Salesorderdetails

('O20015','P1008',15),

('O20015','P1007',10),

('O20016','P1007',20);

('O20016','P1003',5);

- Using Joining table to combine rows from more tables. (NATURAL JOIN, INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN, SEFT JOIN)

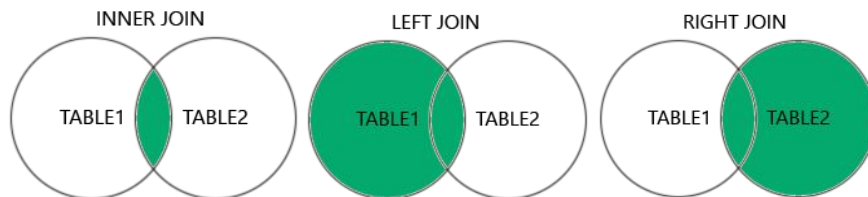
Supported Types of Joins in MySQL

INNER JOIN: Returns records that have matching values in both tables

LEFT JOIN: Returns all records from the left table, and the matched records from the right table

RIGHT JOIN: Returns all records from the right table, and the matched records from the left table

CROSS JOIN: Returns all records from both tables



- Display the clients (name) who lives in same city.

```

184  -- 1.  Display the clients (name) who lives in same city.
185  •  SELECT
186      c.client_name, s.Salesman_Name, c.city, s.city
187  FROM
188      clients c
189      INNER JOIN
190      salesman s ON c.city = s.city;
191  -- 2.  Display city, the client names and salesman names who

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
client_name	Salesman_Name	city	city
Mai Xuan	Tien	Dai An	Dai An
Le Xuan	Hoa	Thu Dau Mot	Thu Dau Mot
Le Xuan	Deb	Thu Dau Mot	Thu Dau Mot
Le Xuan	Khoa	Thu Dau Mot	Thu Dau Mot
Trinh Huu	Quang	Da Lat	Da Lat
Trinh Huu	Tin	Da Lat	Da Lat
Tran Tuan	Hoa	Thu Dau Mot	Thu Dau Mot
Tran Tuan	Deb	Thu Dau Mot	Thu Dau Mot
Tran Tuan	Khoa	Thu Dau Mot	Thu Dau Mot
Ho Nhu	Phat	Hanoi	Hanoi
Tran Hai	Huu	Ho Chi Minh	Ho Chi Minh
Nguyen Thanh	Tien	Dai An	Dai An
Nguyen Su	Quang	Da Lat	Da Lat

2. Display city, the client names and salesman names who are lives in “Thu Dau Mot” city.

```
192 • SELECT DISTINCT
193     c.client_name, c.city, s.Salesman_Name, s.city
194 FROM
195     clients c
196     INNER JOIN
197     salesman s ON c.city like 'Thu Dau Mot' and s.city like 'Thu Dau Mot'
198 -- 3. Display client name, client number, order number
199 • SELECT
```

< Result Grid Filter Rows: Export: Wrap Cell Content:

	client_name	city	Salesman_Name	city
▶	Le Xuan	Thu Dau Mot	Hoa	Thu Dau Mot
	Le Xuan	Thu Dau Mot	Deb	Thu Dau Mot
	Le Xuan	Thu Dau Mot	Khoa	Thu Dau Mot
	Tran Tuan	Thu Dau Mot	Hoa	Thu Dau Mot
	Tran Tuan	Thu Dau Mot	Deb	Thu Dau Mot
	Tran Tuan	Thu Dau Mot	Khoa	Thu Dau Mot




3. Display client name, client number, order number, salesman number, and product number for each order.

```

198 -- 3. Display client name, client number, order number, salesman n
199 • SELECT
200     c.Client_Name,
201     c.Client_Number,
202     so.Order_Number,
203     so.Salesman_Number,
204     sod.Product_Number
205 FROM
206     Clients c
207     JOIN Salesorder so ON c.Client_Number = so.Client_Number
208     JOIN Salesorderdetails sod ON so.Order_Number = sod.Order_Number;
209 -- 4. Find each order (client_number, client_name, order_number) p

```

<

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	Client_Name	Client_Number	Order_Number	Salesman_Number	Product_Number
▶	Mai Xuan	C101	O20001	S003	P1001
	Mai Xuan	C101	O20001	S003	P1002
	Mai Xuan	C101	O20005	S003	P1001
	Mai Xuan	C101	O20005	S003	P1008
	Mai Xuan	C101	O20005	S003	P1002
	Mai Xuan	C101	O20009	S004	P1008
	Le Xuan	C102	O20002	S003	P1007
(1	Le Xuan	C102	O20008	S006	P1004
(1	Trinh Huu	C103	O20003	S002	P1003
	Tran Tuan	C104	O20004	S003	P1004
	Ho Nhu	C105	O20006	S005	P1002
	Tran Hai	C106	O20007	S001	P1005
	Tran Hai	C106	O20010	S006	P1006

Result 92 ▾

4. Find each order (client_number, client_name, order_number) placed by each client.

```
209 -- 4. Find each order (client_number, client_name, or
210 • SELECT
211     c.Client_Number, c.Client_Name, so.Order_Number
212 FROM
213     salesorder so
214     INNER JOIN
215     clients c ON c.Client_Number = so.Client_Number;
216 -- 5. Display the details of clients (client_number,
217 • SELECT
```

< Result Grid Filter Rows: Export: Wrap Cell Contents:

	Client_Number	Client_Name	Order_Number
▶	C101	Mai Xuan	O20001
	C101	Mai Xuan	O20005
	C101	Mai Xuan	O20009
	C102	Le Xuan	O20002
	C102	Le Xuan	O20008
	C103	Trinh Huu	O20003
	C104	Tran Tuan	O20004
[1]	C105	Ho Nhu	O20006
[1]	C106	Tran Hai	O20007
	C106	Tran Hai	O20010
	C107	Nguyen Th...	O20011
	C108	Nguyen Sy	O20012
	C108	Nguyen Sy	O20015

Result 02

5. Display the details of clients (client_number, client_name) and the number of orders which is paid by them.


```

217 • SELECT
218     c.client_number, c.client_name, COUNT(so.Order_Number)
219 FROM
220     clients c
221     JOIN
222     salesorder so ON c.Client_Number = so.Client_Number
223     where Order_Status LIKE 'Successful'
224 GROUP BY client_number , client_name;
225 -- 6. Display the details of clients (client_number, client_

```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	client_number	client_name	COUNT(so.Order_Number)
▶	C101	Mai Xuan	2
	C103	Trinh Huu	1
	C104	Tran Tuan	1
	C106	Tran Hai	2
	C108	Nguyen Sy	2
	C110	Tran Minh	1

6. Display the details of clients (client_number, client_name) who have paid for more than 2 orders.

```

226 • SELECT
227     c.Client_Number, c.Client_Name, COUNT(so.Order_Number) as num_orders
228 FROM
229     clients c
230     JOIN
231     salesorder so ON c.Client_Number = so.Client_Number
232 GROUP BY client_number , client_name
233 having COUNT(so.Order_Number)>2;

```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	Client_Number	Client_Name	num_orders
▶	C101	Mai Xuan	3

7. Display details of clients who have paid for more than 1 order in descending order of client_number.

```

234 -- 7. Display details of clients who have paid for more than 1 order in descending order of
235 • SELECT
236     c.*, COUNT(so.Order_Number) AS num_orders
237 FROM
238     clients c
239     INNER JOIN
240     salesorder so ON c.Client_Number = so.Client_Number
241 GROUP BY c.Client_Number
242 ORDER BY num_orders DESC;
243 -- 8. Find the salesman names who sells more than 20 products.

```

Result Grid Filter Rows: Export: Wrap Cell Content:

	Client_Number	Client_Name	Address	City	Pincode	Province	Amount_Paid	Amount_Due	num_orders
▶	C101	Mai Xuan	Phu Hoa	Dai An	700001	Binh Duong	10000.0000	5000.0000	3
	C102	Le Xuan	Phu Hoa	Thu Dau Mot	700051	Binh Duong	18000.0000	3000.0000	2
	C106	Tran Hai	Phu Hoa	Ho Chi Minh	700002	Ho Chi Minh	7000.0000	1300.0000	2
	C108	Nguyen Sy	Tan An	Da Lat	700032	Lam Dong	15000.0000	1000.0000	2
	C109	Duong Thanh	Phu Hoa	Ho Chi Minh	700011	Ho Chi Minh	12000.0000	8000.0000	2
	C103	Trinh Huu	Phu Loi	Da Lat	700051	Lam Dong	7000.0000	3200.0000	1
	C104	Tran Tuan	Phu Tan	Thu Dau Mot	700080	Binh Duong	8000.0000	0.0000	1
	C105	Ho Nhu	Chanh My	Hanoi	700005	Hanoi	7000.0000	150.0000	1
	C107	Nguyen Thanh	Hoa Phu	Dai An	700023	Binh Duong	8500.0000	7500.0000	1
	C110	Tran Minh	Phu My	Hanoi	700005	Hanoi	9000.0000	1000.0000	1

8. Find the salesman names who sells more than 20 products.





```
244 • select s.Salesman_Name, sum(Target_Achieved) from salesman s
245 join salesorder so on s.Salesman_Number = so.Salesman_Number
246 join salesorderdetails sod on so.Order_Number = sod.Order_Number
247 group by s.Salesman_Name
248 having sum(Target_Achieved) >20;
249 -- 9. Display the client information (client_number, client_name)
250 • SELECT
251 c.Client_Number,
```

< **Result Grid** |   Filter Rows: | Export:  | Wrap Cell Content: 

	Salesman_Name	sum(Target_Achieved)
▶	Huu	105
	Phat	110
	Khoa	210
	Tien	216
	Deb	96
	Tin	165
	Quang	190
	Hoa	150

9. Display the client information (client_number, client_name) and order number of those clients who have order status is cancelled.




```
250 • SELECT
251     c.Client_Number,
252     c.Client_Name,
253     COUNT(so.Order_Number) AS num_orders
254 FROM
255     clients c
256     inner JOIN
257     salesorder so ON c.Client_Number = so.Client_Number
258 WHERE
259     Order_Status LIKE 'Cancelled'
260 GROUP BY c.Client_Number , c.Client_Name;
261 -- 10. Display client name, client number of clients C101 and
```

<	Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Client_Number	Client_Name	num_orders			
▶	C102	Le Xuan	1			
	C101	Mai Xuan	1			
	C107	Nguyen Thanh	1			

10. Display client name, client number of clients C101 and count the number of orders which were received “successful”.

```
262 • SELECT
263     c.Client_Number,
264     c.Client_Name,
265     COUNT(so.Order_Number) AS num_orders
266 FROM
267     clients c
268     inner JOIN
269     salesorder so ON so.Client_Number = 'C101' and c.Client_Number = 'C101'
270 WHERE
271     Order_Status LIKE 'successful'
272 GROUP BY c.Client_Number , c.Client_Name;
273 -- 11. Count the number of clients orders placed for each product.
```

it <

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	Client_Number	Client_Name	num_orders
▶	C101	Mai Xuan	2

11. Count the number of clients orders placed for each product.

```
274 • SELECT
275     p.Product_Name,
276     p.Product_Number,
277     COUNT(sod.Order_Number) AS num_clients
278 FROM
279     product p
280     JOIN
281     salesorderdetails sod ON p.Product_Number = sod.Product_Number
282 GROUP BY p.Product_Name , p.Product_Number;
283 -- 12. Find product numbers that were ordered by more than two clients t
284 • SELECT
285     p.Product_Number, COUNT(so.client_number) AS num_clients
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	Product_Name	Product_Number	num_clients
▶	AC	P1003	2
	Headset	P1008	3
	Keyboard	P1007	4
	Laptop	P1002	4
	Modem	P1004	2
	Mouse	P1006	2
	Pen	P1005	2
(1)	TV	P1001	4

12. Find product numbers that were ordered by more than two clients then order in descending by product number.

```
284 • SELECT
285     p.Product_Number, COUNT(so.client_number) AS num_clients
286 FROM
287     Product p
288     JOIN
289     Salesorderdetails sod ON p.Product_Number = sod.Product_Number
290     JOIN
291     Salesorder so ON sod.Order_Number = so.Order_Number
292 GROUP BY p.Product_Number
293 HAVING num_clients > 2;
```



Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	Product_Number	num_clients
►	P1001	4
	P1002	4
	P1007	4
	P1008	3

b) Using nested query with operator (IN, EXISTS, ANY and ALL)

13. Find the salesman's names who is getting the second highest salary.



```
296 • SELECT
297     Salesman_Name, Salary
298 FROM
299     salesman
300 WHERE
301     Salary = (SELECT DISTINCT
302                Salary
303              FROM
304                salesman
305              ORDER BY Salary DESC
306              LIMIT 1 , 1);
```

< Result Grid   Filter Rows: Export:

	Salesman_Name	Salary
▶	Tin	20000.0000

14. Find the salesman's names who is getting second lowest salary.


```
318 • SELECT
319     Salesman_Name, Salary
320 FROM
321     salesman
322 WHERE
323     Salary = (SELECT DISTINCT
324                Salary
325              FROM
326                salesman
327              ORDER BY Salary
328              LIMIT 1 , 1);
329 -- 15. Write a query to find the name
330 • SELECT
```

< Result Grid   Filter Rows: Export:

	Salesman_Name	Salary
▶	Huu	15000.0000

15. Write a query to find the name and the salary of the salesman who have a higher salary than the salesman whose salesman number is S001.

```
330 • SELECT
331     Salesman_Name, Salary
332 FROM
333     salesman
334 WHERE
335     Salary > (SELECT
336                 Salary
337             FROM
338                 salesman
339             WHERE
340                 Salesman_Number LIKE 'S001');
341 -- 16. Write a query to find the name of all
342 • SELECT
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wra		
	Salesman_Name	Salary
▶	Phat	25000.0000
	Khoa	17500.0000
	Tien	16500.0000
	Tin	20000.0000
	Quang	25000.0000

16. Write a query to find the name of all salesman who sold the product has number: P1002.

```
342 • SELECT
343     sm.Salesman_Name
344 FROM
345     salesman sm
346 WHERE
347     sm.Salesman_Number IN (SELECT
348         so.Salesman_Number
349     FROM
350         salesorder so
351         JOIN
352         salesorderdetails sod ON sod.Order_Number = so.Order_Number
353         JOIN
354         product p ON p.Product_Number = sod.Product_Number
355     WHERE
356         p.Product_Number = 'P1002');
```





Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	Salesman_Name
▶	Khoa
	Deb
	Huu

17. Find the name of the salesman who sold the product to client C108 with delivery status is “delivered”.

```
358 • SELECT
359     sm.Salesman_Name
360 FROM
361     salesman sm
362 WHERE
363     sm.Salesman_Number IN (SELECT
364         Salesman_Number
365     FROM
366         salesorder so
367         JOIN
368         clients c ON c.Client_Number = so.Client_Number
369     WHERE
370         so.Client_Number = 'C108'
371         AND so.Delivery_Status = 'Delivered');
372 -- 18. Display lists the ProductName in ANY records in the sale 0
373 • SELECT
```

<





Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

Salesman_Name

18. Display lists the ProductName in ANY records in the sale Order Details table has Order Quantity equal to 5.

```
373 • SELECT
374     p.Product_Name
375 FROM
376     product p
377     INNER JOIN
378     salesorderdetails sod ON p.Product_Number = sod.Product_Number
379 WHERE
380     sod.Order_Quantity = 5;
```

<

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	Product_Name
▶	TV
	Laptop
	AC

19. Write a query to find the name and number of the salesman who sold pen or TV or laptop.

```
382 • SELECT
383     sm.Salesman_Name, sm.Salesman_Number
384 FROM
385     salesman sm
386 WHERE
387     sm.Salesman_Number IN (SELECT
388         Salesman_Number
389     FROM
390         salesorder sq
391     JOIN
392         salesorderdetails sod ON sod.Order_Number = so.Order_Number
393     JOIN
394         product p ON p.Product_Number = sod.Product_Number
395     WHERE
396         p.Product_Name in ('Pen', 'Tv', 'laptop'));
```

Result Grid

	Salesman_Name	Salesman_Number
►	Khoa	S003
	Deb	S005
	Huu	S001
	Tien	S004
	Tin	S006
*	NULL	NULL

20. Lists the salesman's name sold product with a product price less than 800 and Quantity_On_Hand more than 50.

```
398 • SELECT distinct
399     salesman.Salesman_Name, product.Product_Name
400 FROM
401     salesman
402     INNER JOIN
403     salesorder ON salesman.Salesman_Number = salesorder.Salesman_Number
404     INNER JOIN
405     salesorderdetails ON salesorder.Order_Number = salesorderdetails.Order_Number
406     INNER JOIN
407     product ON product.Product_Number = salesorderdetails.Product_Number
408 WHERE
409     product.Cost_Price < 800
410     AND product.Quantity_On_Hand > 50;
411 -- 21. Write a query to find the name and salary of the salesman whose salary is g
412 • SELECT
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Salesman_Name	Product_Name			
Khoa	Headset			
Tien	Headset			
Quang	Headset			

21. Write a query to find the name and salary of the salesman whose salary is greater than the average salary.

```
412 • SELECT
413     Salesman_Name, Salary
414 FROM
415     salesman
416 WHERE
417     Salary > (SELECT AVG(Salary) as avg_salary FROM salesman);
418 -- 22. Write a query to find the name and Amount Paid of the cl:
419 • SELECT
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Salesman_Name	Salary
▶	Phat	25000.0000
	Tin	20000.0000
	Quang	25000.0000

22. Write a query to find the name and Amount Paid of the clients whose amount paid is greater than the average amount paid.

```
419 • SELECT
420     Client_Name, Amount_Paid
421 FROM
422     clients
423 WHERE
424     Amount_Paid > (SELECT
425                     AVG(Amount_Paid)
426                     FROM
427                     clients);
428 -- II. Additional excersice:
```

Result Grid | Filter Rows: | Export:

	Client_Name	Amount_Paid
▶	Le Xuan	18000.0000
	Nguyen Sy	15000.0000
	Duong Thanh	12000.0000

II. Additional excersice:



23. Find the product price that was sold to Le Xuan.

```
430 • SELECT
431     p.Product_Name, p.Sell_Price
432 FROM
433     product p
434 INNER JOIN
435     salesorderdetails sod ON p.Product_Number = sod.Product_Number
436 INNER JOIN
437     salesorder so ON sod.Order_Number = so.Order_Number
438 INNER JOIN
439     clients c ON so.Client_Number = c.Client_Number
440 INNER JOIN
441     salesman sm ON sm.Salesman_Number = so.Salesman_Number
442 WHERE
443     c.Client_Name LIKE 'Le Xuan';
444 -- 24. Determine the product name, client name and amount due that wa
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Product_Name	Sell_Price			
►	Keyboard	120.0000			
	Modem	250.0000			

24. Determine the product name, client name and amount due that was delivered.

```
445 • SELECT
446     p.Product_Name, c.Client_Name, c.Amount_Due
447 FROM
448     product p
449 INNER JOIN
450     salesorderdetails sod ON p.Product_Number = sod.Product_Number
451 INNER JOIN
452     salesorder so ON sod.Order_Number = so.Order_Number
453 INNER JOIN
454     clients c ON so.Client_Number = c.Client_Number
455 INNER JOIN
456     salesman sm ON sm.Salesman_Number = so.Salesman_Number
457 WHERE
458     so.Delivery_Status = 'Delivered';
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
	Product_Name	Client_Name	Amount_Due
▶	TV	Mai Xuan	5000.0000
	Laptop	Mai Xuan	5000.0000
	Keyboard	Le Xuan	3000.0000
	AC	Trinh Huu	3200.0000
	Modem	Tran Tuan	0.0000
	Pen	Tran Hai	1300.0000
	Mouse	Tran Hai	1300.0000
	TV	Tran Hai	1300.0000

25. Find the salesman's name and their product name which is cancelled.

```
460 • SELECT distinct
461     salesman.Salesman_Name, product.Product_Name
462 FROM
463     salesman
464     INNER JOIN
465     salesorder ON salesman.Salesman_Number = salesorder.Salesman_Number
466     INNER JOIN
467     salesorderdetails ON salesorder.Order_Number = salesorderdetails.Order_Number
468     INNER JOIN
469     product ON product.Product_Number = salesorderdetails.Product_Number
470 WHERE
471     salesorder.Order_Status = 'Cancelled';
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Salesman_Name	Product_Name			
▶	Khoa	Keyboard			
	Khoa	TV			
	Khoa	Headset			
	Khoa	Laptop			
	Deb	Keyboard			

26. Find product names, prices and delivery status for those products purchased by Nguyen Thanh.

```
473 • SELECT
474     p.Product_Name, p.Sell_Price, so.Delivery_Status
475 FROM
476     product p
477 INNER JOIN
478     salesorderdetails sod ON p.Product_Number = sod.Product_Number
479 INNER JOIN
480     salesorder so ON sod.Order_Number = so.Order_Number
481 INNER JOIN
482     clients c ON so.Client_Number = c.Client_Number
483 INNER JOIN
484     salesman sm ON sm.Salesman_Number = so.Salesman_Number
485 WHERE
486     c.Client_Name LIKE 'Nguyen Thanh';
487 -- 27. Display the product name, sell price, salesperson name, delive
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	Product_Name	Sell_Price	Delivery_Status
▶	Keyboard	120.0000	Ready to Ship

27. Display the product name, sell price, salesperson name, delivery status, and order quantity information for each customer.

```

488 • SELECT
489     c.Client_Name, p.Product_Name, p.Sell_Price, sm.Salesman_Name, so.Delivery_Status, sod.Order_Quantity
490 FROM
491     product p
492 INNER JOIN
493     salesorderdetails sod ON p.Product_Number = sod.Product_Number
494 INNER JOIN
495     salesorder so ON sod.Order_Number = so.Order_Number
496 INNER JOIN
497     clients c ON so.Client_Number = c.Client_Number
498 INNER JOIN
499     salesman sm ON sm.Salesman_Number = so.Salesman_Number
500 ORDER BY c.Client_Name;
501 -- 28. Find the names, product names, and order dates of all sales staff whose product order status has been succe

```

Client_Name	Product_Name	Sell_Price	Salesman_Name	Delivery_Status	Order_Quantity
Duong Thanh	Mouse	100.0000	Huu	Ready to Ship	10
Duong Thanh	Keyboard	120.0000	Hoa	Ready to Ship	20
Duong Thanh	AC	400.0000	Hoa	Ready to Ship	5
Ho Nhu	Laptop	1500.0000	Deb	Ready to Ship	5
Le Xuan	Keyboard	120.0000	Khoa	Delivered	10
Le Xuan	Modem	250.0000	Tin	Ready to Ship	8
Mai Xuan	TV	1000.0000	Khoa	Delivered	5
Mai Xuan	Laptop	1500.0000	Khoa	Delivered	4
Mai Xuan	TV	1000.0000	Khoa	On Way	8
Mai Xuan	Headset	50.0000	Khoa	On Way	15
Mai Xuan	Laptop	1500.0000	Khoa	On Way	14

Result 122 x

28. Find the names, product names, and order dates of all sales staff whose product order status has been successful but the items have not yet been delivered to the client.

```

502 • SELECT
503     c.Client_Name, p.Product_Name, so.Order_Date, so.Order_Status, so.Delivery_Status
504 FROM
505     product p
506 INNER JOIN
507     salesorderdetails sod ON p.Product_Number = sod.Product_Number
508 INNER JOIN
509     salesorder so ON sod.Order_Number = so.Order_Number
510 INNER JOIN
511     clients c ON so.Client_Number = c.Client_Number
512 INNER JOIN
513     salesman sm ON sm.Salesman_Number = so.Salesman_Number
514 WHERE
515     so.Order_Status = 'Successful' and not so.Delivery_Status = 'Delivered';
516 -- 29. Find each clients' product which in on the way.



```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Client_Name	Product_Name	Order_Date	Order_Status	Delivery_Status
▶	Mai Xuan	Headset	2022-04-24	Successful	On Way
	Nguyen Sy	Pen	2022-05-12	Successful	On Way
	Nguyen Sy	TV	2022-05-12	Successful	On Way
	Tran Minh	Laptop	2022-05-16	Successful	On Way
	Nguyen Sy	Headset	2022-05-12	Successful	On Way
	Nguyen Sy	Keyboard	2022-05-12	Successful	On Way


29. Find each clients' product which in on the way.

```
517 • SELECT
518     c.Client_Name, p.Product_Name , so.Delivery_Status
519 FROM
520     product p
521 INNER JOIN
522     salesorderdetails sod ON p.Product_Number = sod.Product_Number
523 INNER JOIN
524     salesorder so ON sod.Order_Number = so.Order_Number
525 INNER JOIN
526     clients c ON so.Client_Number = c.Client_Number
527 INNER JOIN
528     salesman sm ON sm.Salesman_Number = so.Salesman_Number
529 WHERE
530     so.Delivery_Status = 'on way';
531 -- 30. Find salary and the salesman's names who is getting the highest salary.
532 • SELECT
```

<			
Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
	Client_Name	Product_Name	Delivery_Status
▶	Mai Xuan	TV	On Way
	Mai Xuan	Headset	On Way
	Mai Xuan	Laptop	On Way
	Mai Xuan	Headset	On Way
	Nguyen Sy	Pen	On Way
	Nguyen Sy	TV	On Way
	Tran Minh	Laptop	On Way
	Nguyen Sy	Headset	On Way
	Nguyen Sy	Keyboard	On Way


30. Find salary and the salesman's names who is getting the highest salary.

```
531 -- 30. Find salary and the salesman's names
532 • SELECT
533     Salesman_Name, Salary
534 FROM
535     salesman
536 ORDER BY Salary DESC
537 LIMIT 1;
538 -- 31. Find salary and the salesman's names
539 • SELECT
540     Salesman_Name, Salary
```

<	
Result Grid	Filter Rows: <input type="text"/>
Export:  W	
Salesman_Name	Salary
▶ Phat	25000.0000





31. Find salary and the salesman's names who is getting second lowest salary.

```
539 • SELECT
540     Salesman_Name, Salary
541 FROM
542     salesman
543 ORDER BY
544     Salary
545 LIMIT 1,1;
546 -- 32. Display lists the ProductName :
```

<	
Result Grid	Filter Rows: <input type="text"/>
Export: 	
Salesman_Name	Salary
▶ Deb	13500.0000

32. Display lists the ProductName in ANY records in the sale Order Details table has Order Quantity more than 9.

```
547 • SELECT
548     p.Product_Name , sod.Order_Quantity
549 FROM
550     product p
551 INNER JOIN
552     salesorderdetails sod ON p.Product_Number = sod.Product_Number
553 WHERE
554     sod.Order_Quantity > 9;
555 -- 33. Find the name of the customer who ordered the same item multi
556 • SELECT distinct
557     c.Client_Name, p.Product_Name
558 FROM
```

< **Result Grid** |   Filter Rows: | Export:  | Wrap Cell Content: 

	Product_Name	Order_Quantity
▶	Keyboard	10
	AC	12
	Headset	15
	Laptop	14
	Mouse	11
	Mouse	10
	Laptop	20
	Headset	15
	Keyboard	10
	Keyboard	20

33. Find the name of the customer who ordered the same item multiple times.

```

569 • SELECT
570     c.Client_Name
571 FROM
572     clients c
573     JOIN
574     (SELECT
575         so.Client_Number, sod.Product_Number
576     FROM
577         salesorder so
578     JOIN salesorderdetails sod ON so.Order_Number = sod.Order_Number
579     GROUP BY so.Client_Number , sod.Product_Number
580     HAVING COUNT(*) > 1) sub ON c.Client_Number = sub.Client_Number;
581 -- 34. Write a query to find the name, number and salary of the salesman
582 • SELECT

```

< Result Grid Filter Rows: Export: Wrap Cell Content:

	Client_Name
▶	Mai Xuan
	Mai Xuan
	Mai Xuan

34. Write a query to find the name, number and salary of the salesmen who earns less than the average salary and works in any of Thu Dau Mot city.

```

582 • SELECT
583     Salesman_Name, Salesman_Number, Salary
584 FROM
585     salesman
586 WHERE
587     City = 'Thu Dau Mot' and Salary < (select avg(Salary) from salesman where City = 'Thu Dau Mot');
588 -- 35. Write a query to find the name, number and salary of the salesmen who earn a salary that is l
589 • SELECT
590     Salesman_Name, Salesman_Number, Salary

```

< Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	Salesman_Name	Salesman_Number	Salary
▶	Deb	S005	13500.0000
	Hoa	S008	13500.0000
*	NULL	NULL	NULL

35. Write a query to find the name, number and salary of the salemans who earn a salary that is higher than the salary of all the salesman have (Order_status = 'Cancelled'). Sort the results of the salary of the lowest to highest.

```

589 • SELECT
590     Salesman_Name, Salesman_Number, Salary
591 FROM
592     salesman
593 WHERE
594     Salary > (SELECT
595         AVG(sm.Salary)
596     FROM
597         salesman sm
598         INNER JOIN
599         salesorder so ON sm.Salesman_Number = so.Salesman_Number
600     WHERE
601         so.Order_Status = 'Cancelled')
602 ORDER BY Salary;
603 -- 36. Write a query to find the 4th maximum salary on the salesman's table.

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content

	Salesman_Name	Salesman_Number	Salary
▶	Tien	S004	16500.0000
	Khoa	S003	17500.0000
	Tin	S006	20000.0000
	Phat	S002	25000.0000
	Quang	S007	25000.0000
•	NULL	NULL	NULL

36. Write a query to find the 4th maximum salary on the salesman's table.

```
604 • SELECT
605     Salesman_Name, Salary
606 FROM
607     salesman
608 ORDER BY Salary DESC
609 LIMIT 4;
```

Result Grid | Filter Rows: Export:

	Salesman_Name	Salary
▶	Phat	25000.0000
	Quang	25000.0000
	Tin	20000.0000
	Khoa	17500.0000

37. Write a query to find the 3th minimum salary in the salesman's table.

```
611 • SELECT
612     Salesman_Name, Salary
613 FROM
614     salesman
615 ORDER BY Salary
616 LIMIT 3;
617
```

Result Grid | Filter Rows:

	Salesman_Name	Salary
▶	Deb	13500.0000
	Hoa	13500.0000
	Huu	15000.0000

Link github: <https://github.com/kaito7love/CSE301.git>