



Practice Assignment 6

Instruction:

** Students are allowing 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.*

I. Use salemanagement database and do following tasks:

A. Add, delete, update value for the columns in table related to calculate new values and constraint.

Note: executing the UPDATE command: In MySQL Workbench if announcement is 1175 error.

1. Go to Edit --> Preferences
2. Click "SQL Editor" tab and uncheck "Safe Updates" check box
3. Query --> Reconnect to Server // logout and then login
4. Now execute your SQL query

Or SET SQL_SAFE_UPDATES = 0;

1. How to check constraint in a table?

Example:

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE  
FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
```

WHERE TABLE_NAME =

'salesman';

```
148 -- lab 6
149 -- A. Add, delete, update value for the columns in table related to calculate new values and constraint.
150 -- 1. How to check constraint in a table?
151 -- use to check all constraint in the table
152 • SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE
153 FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
154 WHERE TABLE_NAME = 'salesman';
155
156 • Select *
157 FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
```

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

	CONSTRAINT_NAME	CONSTRAINT_TYPE
▶	Phone	UNIQUE
	PRIMARY	PRIMARY KEY
	salesman_chk_1	CHECK
	salesman_chk_2	CHECK
	salesman_chk_3	CHECK
	Phone	UNIQUE
	PRIMARY	PRIMARY KEY
	salesman_chk_1	CHECK
	salesman_chk_2	CHECK
	salesman_chk_3	CHECK
	Phone	UNIQUE
	PRIMARY	PRIMARY KEY
	salesman_chk_1	CHECK
	salesman_chk_2	CHECK
	salesman_chk_3	CHECK
	Phone	UNIQUE
	PRIMARY	PRIMARY KEY
	salesman_chk_1	CHECK
	salesman_chk_2	CHECK
	salesman_chk_3	CHECK

2. Create a separate table name as “ProductCost” from “Product” table, which contains the information about product name and its buying price.

```
172
173 -- 2. Create a separate table name as “ProductCost” from “Product” table, which contains the information about product name and its buying price.
174 • CREATE TABLE ProductCost ( -- drop table ProductCost
175     Product_Name VARCHAR(25) NOT NULL UNIQUE,
176     Cost_Price DECIMAL(15,4) NOT NULL,
177     PRIMARY KEY (Product_Name),
178     FOREIGN KEY (Product_Name) REFERENCES Product(Product_Name)
179 );
180 • SHOW CREATE TABLE ProductCost;
181
182 -- 3. Compute the profit percentage for all products. Note: profit = (sell-cost)/cost*100
183 • SELECT Product_Name, Sell_Price, Cost_Price,
184     (Sell_Price - Cost_Price) / Cost_Price * 100 AS profit_percentage
185 FROM Product;
186
187 -- 4. If a salesman exceeded his sales target by more than equal to 75% his remarks should be 'Good'
```

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

Table	Create Table
▶ ProductCost	CREATE TABLE `productcost` (`Product_Nam...

3. Compute the profit percentage for all products. Note: profit = (sell-cost)/cost*100

```

181
182 -- 3. Compute the profit percentage for all products. Note: profit = (sell-cost)/cost*100
183 • SELECT Product_Name,Sell_Price,Cost_Price,
184 (Sell_Price-Cost_Price)/Cost_Price*100 AS profit_percentage
185 FROM Product;
186
187 -- 4. If a salesman exceeded his sales target by more than equal to 75%, his remarks should be 'Good'
188 • SELECT
189     Salesman_Name,
190     Sales_Target

```

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

	Product_Name	Sell_Price	Cost_Price	profit_percentage
▶	TV	1000.0000	800.0000	25.00000000
	Laptop	1500.0000	1100.0000	36.36363636
	AC	400.0000	300.0000	33.33333333
	Modem	250.0000	230.0000	8.69565217
	Pen	12.0000	8.0000	50.00000000
	Mouse	100.0000	105.0000	-4.76190476
	Keyboard	120.0000	90.0000	33.33333333
	Headset	50.0000	40.0000	25.00000000

4. If a salesman exceeded his sales target by more than equal to 75%, his remarks should be 'Good'.

```

186
187 -- 4. If a salesman exceeded his sales target by more than equal to 75%, his remarks should be 'Good'.
188 • SELECT
189     Salesman_Name,
190     Sales_Target,
191     Target_Achieved,
192     IF(Target_Achieved >= Sales_Target * 1.75, 'Good', NULL) AS Remarks
193 FROM Salesman;
194
195 -- 5. If a salesman does not reach more than 75% of his sales objective, he is labeled as 'Average'

```

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

	Salesman_Name	Sales_Target	Target_Achieved	Remarks
▶	Huu	50	35	NULL
	Phat	100	110	NULL
	Khoa	40	30	NULL
	Tien	70	72	NULL
	Deb	60	48	NULL
	Tin	80	55	NULL

5. If a salesman does not reach more than 75% of his sales objective, he is labeled as 'Average'.

```
193 FROM Salesman;
194
195 -- 5. If a salesman does not reach more than 75% of his sales objective, he is labeled as 'Average'.
196 • SELECT
197     Salesman_Name,
198     Sales_Target,
199     Target_Achieved,
200     IF(Target_Achieved <= Sales_Target * 0.75, 'Average', NULL) AS Remarks
201 FROM Salesman;
202
203 -- 6. If a salesman does not meet more than half of his sales objective, he is considered 'Poor'.
204 • SELECT
205     Salesman_Name,
206     Sales_Target,
207     Target_Achieved
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Salesman_Name	Sales_Target	Target_Achieved	Remarks
Huu	50	35	Average
Phat	100	110	NULL
Khoa	40	30	Average
Tien	70	72	NULL
Deb	60	48	NULL
Tin	80	55	Average

6. If a salesman does not meet more than half of his sales objective, he is considered 'Poor'.

```
202
203 -- 6. If a salesman does not meet more than half of his sales objective, he is considered 'Poor'.
204 • SELECT
205     Salesman_Name,
206     Sales_Target,
207     Target_Achieved,
208     IF(Target_Achieved < Sales_Target * 0.50, 'Poor', NULL) AS Remarks
209 FROM Salesman;
210
211 -- 7. Find the total quantity for each product. (Query)
212 • SELECT Product_Number, SUM(Order_Quantity) AS Total_Quantity
213 FROM SalesOrderDetails
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

IA

	Salesman_Name	Sales_Target	Target_Achieved	Remarks
▶	Huu	50	35	NULL
	Phat	100	110	NULL
	Khoa	40	30	NULL
	Tien	70	72	NULL
	Deb	60	48	NULL
	Tin	80	55	NULL

7. Find the total quantity for each product.

(Query)

```
210
211 -- 7. Find the total quantity for each product. (Query)
212 • SELECT Product_Number, SUM(Order_Quantity) AS Total_Quantity
213 FROM SalesOrderDetails
214 GROUP BY Product_Number;
215
216 -- 8. Add a new column and find the total quantity for each product.
217 • ALTER TABLE Product
218 ADD COLUMN Total_Quantity INT;
219 • UPDATE Product p
220 SET p.Total_Quantity = /
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
	Product_Number	Total_Quantity	
▶	P1001	24	
	P1002	43	
	P1003	12	
	P1004	11	
	P1005	9	
	P1006	21	
	P1007	16	
	P1008	17	

8. Add a new column and find the total quantity for each

```

215
216 -- 8. Add a new column and find the total quantity for each product.
217 • ALTER TABLE Product
218   ADD COLUMN Total_Quantity INT;
219 • UPDATE Product p
220   SET p.Total_Quantity = (
221     SELECT SUM(Order_Quantity)
222     FROM SalesOrderDetails sod
223     WHERE sod.Product_Number = p.Product_Number
224     GROUP BY sod.Product_Number
225   );
226 • select Total_Quantity from Product;
227

```

Result Grid

Total_Quantity
24
43
12
11
9
21
16
17

product.

9. If the Quantity on hand for each product is more than 10, change the discount rate to 10 otherwise set to 5.

```

-- 9. If the Quantity on hand for each product is more than 10, change the discount rate to 10 otherwise set to 5
• ALTER TABLE Product ADD COLUMN Discount_Rate INT DEFAULT 0;
• UPDATE Product
  SET Discount_Rate = IF(Quantity_On_Hand > 10, 10, 5);

```

52 01:25:20 UPDATE Product SET Discount_Rate = IF(Quantity_On_Hand > 10, 10, 5)

10. If the Quantity on hand for each product is more than equal to 20, change the discount rate to 10, if it is between 10 and 20 then change to 5, if it is more than 5 then change to 3 otherwise set to 0.

```

-- 10. If the Quantity on hand for each product is more than equal to 20, change the discount rate to 10, if it is between 10 and 20 then change to 5, if it is more than 5 then change to 3 otherwise set to 0.
• UPDATE Product
  SET Discount_Rate =
    IF(Quantity_On_Hand >= 20, 10,
      IF(Quantity_On_Hand > 10, 5,
        IF(Quantity_On_Hand > 5, 3, 0)
      )
    );

```

56 01:26:42 UPDATE Product SET Discount_Rate = IF(Quantity_On_Hand >= 20, 10, IF(Quantity_On_Hand > 10, 5, IF(Quantity_On_Hand > 5, 3, 0)))

4 row(s) affected Rows matched: 8 Changed: 4 Warnings: 0

11. The first number of pin code in the client table should start with

7.

```
246
247 -- 11. The first number of pin code in the client table should start with 7.
248 • SELECT * FROM Clients WHERE Pincode LIKE '7%'; -- SELECT * FROM Clients WHERE Pincode LIKE '7%';
249 • UPDATE Clients
250 SET Pincode = 700000
251 WHERE Pincode NOT LIKE '7%';
252
253
254 -- B. Creating and using view:
255 -- 12. Create a view name as clients view that shows all customers information from Thu Dau Mot
```

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

12.

B. Creating and using view:

Statements:

1. Creating a view

```
CREATE [OR REPLACE] VIEW view_name
AS
    select-statement;
```

OR REPLACE - added to overwrite the old view with the same name if applicable.

2. Changing a view

```
ALTER VIEW view_name AS select_statement;
```

3. Renaming a view

```
RENAME TABLE view_name TO new_view_name;
```

4. Deleting a view

```
DROP VIEW [IF EXISTS] view_name;
```



```

256 -- 12.
257 -- Create a View
258 • CREATE VIEW view_name AS
259 SELECT * FROM table_name;
260 -- 2.Changing a view
261 • ALTER VIEW view_name AS SELECT column1, column2
262 FROM table_name;
263 -- 3.Renaming a view
264 • RENAME TABLE view_name TO new_view_name;
265 -- Deleting a view
266 • DROP VIEW IF EXISTS view_name;
267

```

13. Creates a view name as clients_view that shows all customers information from Thu Dau

Mot.

```

268
269 -- 13. Creates a view name as clients_view that shows all customers information from Thu Dau Mot.
270 • CREATE VIEW clients_view AS -- DROP VIEW clients_view;
271 SELECT * FROM Clients
272 WHERE City = 'Thu Dau Mot';
273 • SELECT * FROM clients_view ;
274
275 -- 14. Drop the "client_view".
276 • DROP VIEW clients_view;
277

```

Result Grid								
Filter Rows: <input type="text"/> Export: Wrap Cell Content:								
	Client_Number	Client_Name	Address	City	Pincode	Province	Amount_Paid	Amount_Due
▶	C102	Le Xuan	Phu Hoa	Thu Dau Mot	700051	Binh Duong	18000.0000	3000.0000
	C104	Tran Tuan	Phu Tan	Thu Dau Mot	700080	Binh Duong	8000.0000	0.0000

```

274
275 -- 14. Drop the "client_view".
276 • DROP VIEW clients_view;
277

```

14. Drop the "client_view". 277

74 01:41:27 DROP VIEW clients_view

15. Creates a view name as clients_order that shows all clients and their order details from Thu Dau Mot.

```
278
279 -- 15. Creates a view name as clients_order that shows all clients and their order details from Thu Dau Mot.
280 • CREATE VIEW clients_order AS -- DROP VIEW clients_order;
281 SELECT c.Client_Number, c.Client_Name, so.Order_Number, so.Order_Date, so.Order_Status
282 FROM Clients c
283 JOIN SalesOrder so ON c.Client_Number = so.Client_Number
284 WHERE c.City = 'Thu Dau Mot';
285 • SELECT * FROM clients_order;
286
287
288
289 -- 16. Creates a view that selects every product in the "Products" table with a sell price higher than the average se
290 • CREATE VIEW high_price_products AS -- DROP VIEW high_price_products;
```

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

	Client_Number	Client_Name	Order_Number	Order_Date	Order_Status
▶	C102	Le Xuan	O20002	2022-01-25	Cancelled
	C102	Le Xuan	O20008	2022-04-16	In Process
	C104	Tran Tuan	O20004	2022-02-10	Successful

16. Creates a view that selects every product in the "Products" table with a sell price higher than the average sell price.

```
288
289 -- 16. Creates a view that selects every product in the "Products" table with a sell price higher than the average sell price.
290 • CREATE VIEW high_price_products AS -- DROP VIEW high_price_products;
291 SELECT * FROM Product
292 WHERE Sell_Price > (SELECT AVG(Sell_Price) FROM Product);
293 • SELECT * FROM high_price_products;
294
295 -- 17. Creates a view name as salesman_view that show all salesman information and products (product names, product price, quantity o
296 • CREATE VIEW salesman_view AS -- DROP VIEW salesman_view;
```

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

	Product_Number	Product_Name	Quantity_On_Hand	Quantity_Sell	Sell_Price	Cost_Price	Total_Quantity	Discount_Rate
▶	P1001	TV	10	30	1000.0000	800.0000	24	3
	P1002	Laptop	12	25	1500.0000	1100.0000	43	5

17. Creates a view name as salesman_view that show all salesman information and products (product names, product price, quantity order) were sold by

them.

```
294
295 -- 17. Creates a view name as salesman_view that show all salesman information and products (product names, product price, quantity order) were sold by them.
296 • CREATE VIEW salesman_view AS -- DROP VIEW salesman_view;
297 SELECT s.*, p.Product_Name, p.Sell_Price, sod.Order_Quantity
298 FROM Salesman s
299 JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
300 JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
301 JOIN Product p ON sod.Product_Number = p.Product_Number;
302 • SELECT * FROM salesman_view;
303
304 -- 18. Creates a view name as sale_view that show all salesman information and product (product names, product price, quantity order) were sold by them with order_
305 • CREATE VIEW sale_view AS -- DROP VIEW sale_view;
```

Result Grid

	Salesman_Number	Salesman_Name	Address	City	Pincode	Province	Salary	Sales_Target	Target_Achieved	Phone	Product_Name	Sell_Price	Order_Quantity
▶	S001	Huu	Phu Tan	Ho Chi Minh	700002	Ho Chi Minh	15000.0000	50	35	0902361123	Pen	12.0000	6
	S001	Huu	Phu Tan	Ho Chi Minh	700002	Ho Chi Minh	15000.0000	50	35	0902361123	Mouse	100.0000	10
	S001	Huu	Phu Tan	Ho Chi Minh	700002	Ho Chi Minh	15000.0000	50	35	0902361123	Laptop	1500.0000	20
	S002	Phat	Tan An	Hanoi	700005	Hanoi	25000.0000	100	110	0903216542	AC	400.0000	12
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	TV	1000.0000	5
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	Laptop	1500.0000	4
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	Keyboard	120.0000	10
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	Modem	250.0000	3
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	TV	1000.0000	8
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	Headset	50.0000	15
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	Laptop	1500.0000	14
	S004	Tien	Phu Hoa	Dai An	700023	Binh Duong	16500.0000	70	72	0908654723	Headset	50.0000	2
	S004	Tien	Phu Hoa	Dai An	700023	Binh Duong	16500.0000	70	72	0908654723	Pen	12.0000	3
	S004	Tien	Phu Hoa	Dai An	700023	Binh Duong	16500.0000	70	72	0908654723	TV	1000.0000	2
	S005	Deb	Hoa Phu	Thu Dau Mot	700051	Binh Duong	13500.0000	60	48	0903213659	Laptop	1500.0000	5
	S005	Deb	Hoa Phu	Thu Dau Mot	700051	Binh Duong	13500.0000	60	48	0903213659	Keyboard	120.0000	6
	S006	Tin	Chanh My	Da Lat	700032	Lam Dong	20000.0000	80	55	0907853497	Modem	250.0000	8
	S006	Tin	Chanh My	Da Lat	700032	Lam Dong	20000.0000	80	55	0907853497	Mouse	100.0000	11
	S006	Tin	Chanh My	Da Lat	700032	Lam Dong	20000.0000	80	55	0907853497	TV	1000.0000	9

18. Creates a view name as sale_view that show all salesman information and product (product names, product price, quantity order) were sold by them with order_status = 'Successful'.

```
303
304 -- 18. Creates a view name as sale_view that show all salesman information and product (product names, product price, quantity order) were sold by them with order_status = 'Successful'.
305 • CREATE VIEW sale_view AS -- DROP VIEW sale_view;
306 SELECT s.*, p.Product_Name, p.Sell_Price, sod.Order_Quantity
307 FROM Salesman s
308 JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
309 JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
310 JOIN Product p ON sod.Product_Number = p.Product_Number
311 WHERE so.Order_Status = 'Successful';
312 • SELECT * FROM sale_view;
313
314 -- 19. Creates a view name as sale_amount_view that show all salesman information and sum order quantity of product greater than and equal 20 pieces were sold by them with order_status = 'Su
```

Result Grid

	Salesman_Number	Salesman_Name	Address	City	Pincode	Province	Salary	Sales_Target	Target_Achieved	Phone	Product_Name	Sell_Price	Order_Quantity
▶	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	TV	1000.0000	5
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	Laptop	1500.0000	4
	S002	Phat	Tan An	Hanoi	700005	Hanoi	25000.0000	100	110	0903216542	AC	400.0000	12
	S003	Khoa	Phu Hoa	Thu Dau Mot	700051	Binh Duong	17500.0000	40	30	0904589632	Modem	250.0000	3
	S001	Huu	Phu Tan	Ho Chi Minh	700002	Ho Chi Minh	15000.0000	50	35	0902361123	Pen	12.0000	6
	S004	Tien	Phu Hoa	Dai An	700023	Binh Duong	16500.0000	70	72	0908654723	Headset	50.0000	2
	S006	Tin	Chanh My	Da Lat	700032	Lam Dong	20000.0000	80	55	0907853497	Mouse	100.0000	11
	S006	Tin	Chanh My	Da Lat	700032	Lam Dong	20000.0000	80	55	0907853497	TV	1000.0000	9
	S004	Tien	Phu Hoa	Dai An	700023	Binh Duong	16500.0000	70	72	0908654723	Pen	12.0000	3
	S004	Tien	Phu Hoa	Dai An	700023	Binh Duong	16500.0000	70	72	0908654723	TV	1000.0000	2
	S001	Huu	Phu Tan	Ho Chi Minh	700002	Ho Chi Minh	15000.0000	50	35	0902361123	Laptop	1500.0000	20

19. Creates a view name as sale_amount_view that show all salesman information and sum order quantity of product greater than and equal 20 pieces were sold by them with order_status =

'Successful'.

```
314 -- 19. Creates a view name as sale_amount_view that show all salesman information and sum order quantity of product greater than and equal 20 pieces were sold by them with order_status = 'Successful'
315 • CREATE VIEW sale_amount_view AS -- DROP VIEW sale_amount_view;
316 SELECT s.*, SUM(sod.Order_Quantity) AS Total_Quantity
317 FROM Salesman s
318 JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
319 JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
320 JOIN Product p ON sod.Product_Number = p.Product_Number
321 WHERE so.Order_Status = 'Successful'
322 GROUP BY s.Salesman_Number
323 HAVING Total_Quantity >= 20;
324 • SELECT * FROM sale_amount_view;
325
326
327
328 • SELECT * FROM clients_view ;
329 • SELECT * FROM clients_order;
```

Result Grid

	Salesman_Number	Salesman_Name	Address	City	Pincode	Province	Salary	Sales_Target	Target_Achieved	Phone	Total_Quantity
▶	S001	Huu	Phu Tan	Ho Chi Minh	700002	Ho Chi Minh	15000.0000	50	35	0902361123	26
	S006	Tin	Chanh My	Da Lat	700032	Lam Dong	20000.0000	80	55	0907853497	20

II. Additional assignments about Constraint

20. Amount paid and amount due should not be negative when you are inserting the data.

```
-- Extra assignment
-- 20.Amount paid and amount due should not be negative when you are inserting the data.
• SHOW COLUMNS FROM SalesOrder;
• ALTER TABLE SalesOrder
  ADD CONSTRAINT chk_amount_paid_due
  CHECK (Amount_Paid >= 0 AND Amount_Due >= 0); -- avoid negative
```

21. Do not enforce the check constraint for pincode.

```
343 -- 21.Do not enforce the check constraint for pincode.
344 • ALTER TABLE Clients
345 DROP CHECK chk_pincode; -- if there is already an created constraint for pincode in client
346
```

22. How to alter a check constraint enforcement

```
7 -- 22.How to alter a check constraint enforcement state?
3 -- enable
3 • ALTER TABLE Clients
state? • ENABLE KEYS;
```

23. Remove the constraint from

```
56 -- 23.Remove the constraint from pincode; ( same as 20)
57 • ALTER TABLE Clients
pincode; 58 DROP CHECK chk_pincode;
```

24. The sell price and cost price should be

```
360      -- 24.The sell price and cost price should be unique.
361 •    ALTER TABLE Product
362      ADD CONSTRAINT unique_price UNIQUE (Sell_Price, Cost_Price);
unique.
```

25. The sell price and cost price should not be

```
4      -- 25.The sell price and cost price should not be unique. (change above code)
5 •    ALTER TABLE Product
6      DROP INDEX unique_price;
unique.
```

26. Update the delivery status to “Delivered” for the product number

```
8      -- 26.Update the delivery status to “Delivered” for the product number P1007.
9 •    UPDATE SalesOrder
10     SET Delivery_Status = 'Delivered'
11     WHERE Product_Number = 'P1007';
P1007.
```

27. Change address and city to ‘Phu Hoa’ and ‘Thu Dau Mot’ where client number is C104.

```
72
73      -- 27.Change address and city to ‘Phu Hoa’ and ‘Thu Dau Mot’ where client number is C104.
74 •    UPDATE Clients
75     SET Address = 'Phu Hoa', City = 'Thu Dau Mot'
76     WHERE Client_Number = 'C104';
77
```

28. Add a new column to “Product” table named as “Exp_Date”, data type is Date.

```
378      -- 28.Add a new column to “Product” table named as “Exp_Date”, data type is Date.
379 •    ALTER TABLE Product
380     ADD COLUMN Exp_Date DATE;
381
```

29. Add a new column to “Clients” table named as “Phone”, data type is varchar and size is 15.

```
81
82      -- 29.Add a new column to “Clients” table named as “Phone”, data type is varchar and size is 15.
83 •    ALTER TABLE Clients
84     ADD COLUMN Phone VARCHAR(15);
85
```


30. Update remarks as “Good” for all

```
-- 30.Update remarks as “Good” for all salesman.
UPDATE Salesman
SET Remarks = 'Good';
```

salesman.

31. Change remarks to "bad" whose salesman number is

```
389
390 -- 31.Change remarks to "bad" whose salesman number is "S004".
391 • UPDATE Salesman
392 SET Remarks = 'Bad'
393 WHERE Salesman_Number = 'S004';
394
```

"S004".

32. Modify the data type of “Phone” in “Clients” table with varchar from size 15 to size is 10.

```
5 -- 32.Modify the data type of “Phone” in “Clients” table with varchar from size 15 to size is 10.
6 • ALTER TABLE Clients
7 MODIFY Phone VARCHAR(10);
8
```

33. Delete the “Phone” column from “Clients”

```
398
399 -- 33.Delete the “Phone” column from “Clients” table.
400 • ALTER TABLE Clients
401 DROP COLUMN Phone;
402
```

table.

34. alter table Clients drop column

```
402
403 -- 34.alter table Clients drop column Phone; (same as 32)
404 • ALTER TABLE Clients
405 DROP COLUMN Phone;
406
407
```

Phone;

35. Change the sell price of Mouse to

```
406
407 -- 35.Change the sell price of Mouse to 120.
408 • UPDATE Product
409 SET Sell_Price = 120
410 WHERE Product_Name = 'Mouse';
411
```

120.

36. Change the city of client number C104 to “Ben

```
2      -- 36.Change the city of client number C104 to “Ben Cat”.
3 •    UPDATE Clients
4      SET City = 'Ben Cat'
5      WHERE Client_Number = 'C104';
```

Cat”.

37. If Order_Quantity greater than 5, then 10% discount. If Order_Quantity greater than 10, then 15% discount. Otherwise, no discount.

```
416
417      -- 37.If Order_Quantity greater than 5, then 10% discount. If Order_Quantity greater than 10, then 15% discount. Otherwise, no discount
418 •    UPDATE SalesOrderDetails
419      SET Discount = IF(Order_Quantity > 10, 15, IF(Order_Quantity > 5, 10, 0));
420
```