1. Write in SQL to return all the products from 'products' table from ‘sql\_store’ database. the columns should be:
   1. name
   2. unit price
   3. new price (unit\_price increased by 10%)
2. Write in SQL to return all the order\_id from 'orders' table from ‘sql\_store’ database, which are ordered in 2017
3. Write in SQL to return the order\_id from 'order\_items' table from ‘sql\_store’ database, whose total price is greater than 30 and less than 50.
4. Write in SQL to return the order\_id from 'order\_items' table from ‘sql\_store’ database, whose total price is greater than 30 or even.
5. Write in SQL to return product\_id from 'products' table from ‘sql\_store’ whose quantity in stocks are equal to 49 or 38
6. Write in SQL to return customers' first\_name from 'customers' table from ‘sql\_store’ database who born between 1-Jan-1990 and 1-Jan-2000
7. Write in SQL to fetch the customer details from 'customers' table from ‘sql\_store’ database, whose last name starts with 'b' or 'B'
8. Write in SQL to print customer details from 'customers' table from ‘sql\_store’ whose addresses contains 'TRAIL' or 'AVENUE'.
9. Write in SQL to print customer details from 'customers' table from ‘sql\_store’ whose phone numbers end with 9
10. Write in SQL to print customer details from 'customers' table from ‘sql\_store’ whose addresses contains 'TRAIL' or 'AVENUE' using REGULAR EXPRESSIONS
11. Get the customers from 'customers' table from ‘sql\_store’ database whose:
    1. first\_names are ELKA or AMBUR
    2. last\_names end with EY or ON
    3. last\_names start with MY or contains SE
    4. last\_names contain B followed by R or U
12. Write in SQL to find the order list which are not shipped. (table: orders, database: ‘sql\_store’
13. Write in SQL to find the product\_id, quantity and unit\_price for order\_id = 2. Sort the table by the total price (i.e. quantity\*unit\_price). (table: order\_items, database: sql\_store)
14. Write in SQL to find top 3 loyal customers from ‘customers’ table from sql\_store database. (**note:** loyal customer means, customer with more points)
15. Join two tables ‘order\_items’ and ‘products’ from sql\_store database and find a table with order\_id, product\_id, product\_name, quantity and unit\_price.
16. From sql\_hr database, ‘employees’ table fetch all employees firstname and their manager’s name.
17. Use sql\_invoicing databse, and join client, payments, and payment\_methods table to fetch a table having three columns: client\_id, firstname and payment method name.
18. Use sql\_store database and ‘products’ and ‘order\_items’ table. Write in SQL to fetch a table with columns: product\_id, name and quantity. Even if the products are not ordered, they should be listed in the table with quantity as ‘null’.
19. Use sql\_store database and use ‘customers’, ’orders’, ’order\_statuses’, ’shippers’ table, and fetch a table with 5 columns describing order date, order id, firstname of the customer, shipper name and shipping status.
20. Use sql\_invoicing database and use ‘clients’, ‘invoices’, ‘payment\_methods’, and ‘payments’ table and fetch a table with columns with description: payment date, client name, amount paid, and payment method name.
21. Join ‘shippers’ and ‘products’ table by cross join with implicit syntax.
22. Join ‘shippers’ and ‘products’ table by cross join with explicit syntax.
23. Use sql\_store data base and ‘customers’ table and fetch a table with columns: customer’s firstname, points and tier. The tier rules are given below:

|  |  |
| --- | --- |
| **Points** | **Tier** |
| <2000 | Bronze |
| >=2000 & <=3000 | Silver |
| >3000 | Gold |

1. Write in SQL to enter a new set of details of yours in the ‘customers’ table from sql\_store database.
2. Write in SQL to enter three new shipper names in ‘shippers’ table from sql\_store database.
3. Use sql\_store database and ‘customers’ table. Update the points by adding 50 points in the database.
4. Update the ‘order’ tables’ ‘comments’ column with ‘Gold Customer’, if the client has more than 3000 points.