# Setup the Bike Stores database in your SQL developer, then:

**1) Display a list of all sales orders with staff id is 9.**

**2) Name all the customers who live in New York and provide a phone number.**

**3) List all the customers names who ordered from store\_id = 2.**

**4) List all staff names with discount > 0.05 Note: required reference to**

**multiple tables.**

**5) Display the names of the customers who ordered from Baldwin Bikes and Santa Cruz Bikes stores.**

**6) Display the item ID and name whose discount is more than 0.05 .**

**7) Write the join query to produce the following result set**

**(note here: first\_name and last\_names of customers)**

**A screenshot of a computer

Description automatically generated**

**8) Write the join query to produce the following result set.**

**(notex here: first\_name and last\_nanes of staff)**

**A screenshot of a computer

Description automatically generated**

**9 ) Write the join query to produce the following result set.**

**(note here : first name and last names of staff)**

**A screenshot of a computer

Description automatically generated**

**10) Write the join query to produce the following result set.**

**(note here : THREE TABLE JOIN )**

**A screenshot of a table

Description automatically generated**

**11) How many total serving customer does BikeStore has ?**

**12) How many total orders are there ?**

**13) Which store has the highest number of sales ?**

**14) Which store the sales was highest and for which month ?**

**15) How many orders each customer has placed (give me top 10 customers) ?**

**16) Which are the TOP 3 selling product ?**

**17) Which was the first and last order placed by the customer who has placed the maximum number of orders ?**

**18) For every customer , which is the cheapest product and the most cost product which the customer has bought ?**

**19) Which product has ordered more than 100 times ?**

**20) The query retrieves order\_date information ( return results are unique ), sorted in ascending order.**

**21) Query brand\_id information and category\_id ( return results are unique ).**

**22) Write a query to get all employee information with store\_id equal to 1 and manager\_id equal to 2, sorted in ascending order by first\_name.**

**23) Write a query that gets all product information with brand\_id equal to 1 or 9, and has a price between 199.99 and 499.99.**

**24) Write a query that lists the names of the 5 products with the highest price provided that the product has a model\_year equal to 2018.**

**25) Write a query to get all customer information (customers) with first\_name has the end next character is 't' and zip\_code starts with '11', sorted by first\_name.**

**26) Write a query to get all product information has a price equal to 999.99 or 1999.99 or 2999.99.**

**27) Write a query that returns the total number of products whose names start with 'Trek' and priced from 199.99 to 999.99.**

**28) Write a query that returns product name, total price and total quantity of products for each product with the keyword 'Ladies' in the product name.**

**29) Write a query to output information about orders that have a total net value greater than 20,000 on the sales.order\_items table, know the net\_ value calculated by the formula ( quantity \* list\_price \* (1 - discount) )**

**30) Write a query to get product information (product\_name), order code (order\_id) with that product (if any), number of products (quantity) and daily orders transacted (order\_date)**

**31) Write a query to get brand information (brand\_name) and average price (average\_list\_price) for all products with model\_year of 2018.**

**32) Write a query to get information about order code (order\_id), customer name (customer\_name), store name (store\_name), total product quantity (total\_quantity) and total order value (total\_net\_value) knowing order value (net\_value) calculated by the formula quantity \* list\_price \* (1 - discount)**

**33) Write a query to get information about products that have not been sold at any stores or are out of stock (quantity = 0), results should return store name and product name.**

**-----------------------------------------**

**Extra Questions:**

**Query 1:**

Prepare a query result of customers with their first and last names, email, phone, and

state if the customer has no value in the field ‘phone’. The result set should show a

combined first and last name labeled as Full Customer Name and it should be sorted by

state. Use comments to describe this query result.

**Query 2:**

What are the list prices, product names and model years of the top 10 most expensive

products (according to their list price)?

**Query 3:**

List any product information available if the product has the word ‘townie’ anywhere in

the product name.

**Query 4:**

Create a query result that shows the Order Number, Required Date, Shipped Date and a new column that shows the difference between the required date and the shipped date. Only show instances where an order was shipped after the required date.

**Query 5:**

Create a query result that shows the states Jeff’s stores are located in.

**Query 6:**

List any March orders that are not from stores 1 and 3. Include Order ID, Order\_Date

and Store\_ID in the result set. Ensure that the date is formatted as MM-DD-YYYY.

**Query 7:**

Which customers are from New York and purchased an electric bike. Use

correlation/alias names in your joins.

**Query 8:**

Jeff’s Bike Shop management would like to see a result listing of all staff ids and staff

names if the staff member has ever sold to a customer from zip code 11418.

**Query 9:**

Create a result set that will answer the following business question: What stores have

any products in stock with a very low quantity? Jeff determines “very low quantity” as

being less than 4 products left in stock. Provide enough information in the output to be

a meaningful report to management.

**Query 10:**

Which customers have purchased both of these products from our company?

* Surly Straggler 650b - 2016
* Sun Bicycles Cruz 3 – 2017