Exercise 1

1. Create a Database ‘Learning\_space’
2. Create a table named ‘Science\_curriculum’ with the following properties

(Enrollment\_ID INT, Name VARCHAR, Science\_Marks INT)

Exercise 2

1. Insert the following data into Science\_class using insert into command
2. Import data from csv file ’Student.csv’ attached in resources to Science\_curriculum to insert data of next 8 students

Exercise 3

1. Retrieve all data from the table ‘Science\_curriculum’
2. Retrieve the name of students who have scored more than 65 marks
3. Retrieve all data of students who have scored more than 35 but less than 65 marks
4. Retrieve all other students i.e. who have scored less than or equal to 35 or more than or equal to 65.

Exercise 4

1. Update the marks of student 'Alice Johnson' to 90.
2. Delete the row containing details of student named ‘Robb’
3. Rename the column ‘science\_marks’ to ‘marks\_obtained’.
4. Update the marks of student 'Brian' to 65.

Exercise 5

1. Backup this database into a TAR file
2. Drop the ‘Science\_curriculum’ table
3. Restore from the backup file to get back the deleted table

Exercise 6

In the database Supermart\_DB, find the following

1. List all unique cities located in the northern or eastern regions, using the IN statement to filter the results.
2. Identify all orders where the sales amount falls between 100 and 500, using the BETWEEN operator.
3. Retrieve all customers whose last names have exactly four characters, applying the LIKE operator for the search.

Exercise 7

1. Fetch all orders where the discount value is greater than zero, and sort the results in descending order based on the discount value.
2. Restrict the results of the above query to the top 10 entries.

Exercise 8

1. Calculate the total sum of all sales values
2. Determine the number of customers in the north region who are aged between 20 and 30.
3. Compute the average age of customers located in the east region.
4. Identify the minimum and maximum ages of customers from Philadelphia.

Exercise 9

1. Create a dashboard that displays the following metrics for each product ID:
   * Total sales (in $), sorted in descending order
   * Total sales quantity
   * Number of orders
   * Maximum sales value
   * Minimum sales value
   * Average sales value

Additionally, provide a list of product IDs where the quantity sold exceeds 10.

Exercise 10

1. **Query 1**
   * **Retrieve Data**: Get the Product ID, Product Name, Sales Value, and Profit.
   * **Join Method:** Use an INNER JOIN to combine the product and sales tables based on the Product ID column as the common key.
2. **Query 2**
   * **Customer Sales Summary:** Fetch the Customer Name and City along with the total sales amount for each customer.
   * **Grouping:** Use the GROUP BY clause on Customer ID to calculate the total sales for each customer, and then join this result with the customer table to get the corresponding customer details.

Exercise 11

1. 1. Calculate the total sales for each state using the `customer\_20\_60` and `sales\_2015` tables. Remember to utilize joins and the GROUP BY command.
2. Retrieve data that includes the product ID, product name, category, total sales value for each product, and total quantity sold. (Make sure to use the `sales` and `product` tables.)

Exercise 12

Retrieve the names of all customer IDs from the sales table along with the sales information for each customer

listed in the customer table. Display both customer IDs in separate columns.

Exercise 13

Retrieve the customers from the customer table who have total sales exceeding 1000 and reside in 'New York City,' along with those who live in 'Mesa' and have total sales below 500.

Exercise 14

select c.customer\_name, c.age, sp.\* from

customer as c

right join (select s.\*, p.product\_name, p.category

from sales as s

left join product as p

on s.product\_id = p.product\_id) as sp

on c.customer\_id = sp.customer\_id;

Exercise 15

1. Create a view named “Daily\_Billing” that includes `order\_line`, `product\_id`, `sales`, and discount values for the earliest order date in the sales table.
2. Remove this view.

Exercise 16

1. Determine the maximum length of characters in the product name from the Product table.
2. Retrieve the product name, sub-category, and category from the Product table, along with an additional column called “product\_details” that contains a concatenated string of the product name, sub-category, and category.
3. Analyze the product\_id column and extract its three components into three separate columns.
4. List the product names as a comma-separated string where the sub-category is either Chairs or Tables.

Exercise 17

1. You’re hosting a lottery for your customers. Select a list of 5 lucky customers from the customer table using a random function.
2. Since you cannot charge customers in fractional amounts, for a sales value of 1.63, you would charge either 1 or 2. In this case, calculate:
   * The total sales revenue if you always charge the lower integer value.
   * The total sales revenue if you always charge the higher integer value.
   * The total sales revenue if you round the sales amount.

Exercise 18

1. Determine the current age of "Batman," who was born on April 6, 1939, in years, months, and days.
2. Examine the monthly sales for the chair sub-category. Do you notice any seasonal trends in the sales of this sub-category?

Exercise 19

1. Retrieve all customers whose first name and last name are both 5 characters long, and whose last name starts with the letters "a," "b," "c," or "d.“
2. Create a table named "zipcode" and insert the specified data into it.

|  |
| --- |
| PIN/ZIP codes |
| 234432 |
| 23345 |
| sdfe4 |
| 123&3 |
| 67424 |
| 7895432 |
| 12312 |

Retrieve the valid zip codes from this table, which consist of either 5 or 6 numeric characters.

Exercise 20

1. Who are the top customers by total sales in each state?
2. What are the top 5 sub-categories based on total sales?"