**Beginner Level (Assignments 1–8)**

1. **Basic Selection and Filtering**
   * Write a query to display all customer details (SELECT, FROM, SELECT \*).
   * Filter customers to show only those who live in "California" (WHERE, <>).
2. **Sorting and Limiting Results**
   * Display the top 5 most expensive products (SELECT, TOP, ORDER BY).
   * Retrieve the top 10% of products by price (PERCENT).
3. **Using Aliases and DISTINCT**
   * Retrieve a list of unique product categories and alias the column as "Category Type" (DISTINCT, AS).
4. **Combining Criteria**
   * Retrieve all customers from either "Texas" or "Florida" (WHERE, IN).
   * Filter customers who spent more than $500 and purchased in 2023 (AND, BETWEEN).
5. **NULL Handling**
   * Find all products where the description is missing (IS NULL).
   * List all orders where comments are present (IS NOT NULL).
6. **Pattern Matching**
   * Find customers whose names start with "A" or end with "s" (LIKE, Wildcards).
7. **Basic Aggregations**
   * Calculate the total revenue from the order\_items table (SUM).
   * Find the average price of products and the highest price in each category (AVG, MAX, GROUP BY).
8. **String Functions**
   * Display the first 3 characters of product names (LEFT).
   * Find the position of the word "Bike" in product names (CHARINDEX).

**Intermediate Level (Assignments 9–16)**

1. **Substring and Replace**
   * Extract the last 4 digits of customers’ phone numbers (RIGHT, SUBSTRING).
   * Replace "Bike" with "Cycle" in all product names (REPLACE).
2. **Advanced Filtering**
   * List all orders placed between January 1, 2023, and March 31, 2023 (BETWEEN).
   * Exclude all customers who live in "Nevada" or "Alaska" (<>, NOT IN).
3. **Combining Data from Two Tables**
   * Retrieve customer names along with their order details (INNER JOIN).
   * Show products and their associated categories (LEFT JOIN).
4. **Aggregation with HAVING**
   * Display categories with total sales exceeding $10,000 (GROUP BY, HAVING).
   * Find stores with more than 50 orders placed (COUNT, HAVING).
5. **Advanced Sorting**
   * List products sorted by the length of their names (ORDER BY Length of Alphabets).
   * Sort orders based on the second column in the result (ORDER BY with Column Indexes).
6. **DML (Data Manipulation Language)**
   * Insert a new customer into the customers table.
   * Update a product's price by increasing it by 10%.
   * Delete all records from the stocks table where quantity is 0.
7. **Pagination**
   * Retrieve the 6th to 10th most expensive products (OFFSET, FETCH).
8. **Combining String Functions**
   * Find all customers whose phone numbers contain "123" and extract only the digits before it (CHARINDEX, SUBSTRING).

**Advanced Level (Assignments 17–25)**

1. **Subqueries**
   * Find all orders where the total amount exceeds the average order value (SUBQUERY, AVG).
   * List customers who haven’t placed any orders.
2. **Complex Joins**
   * Retrieve all products and their stock details, including stores with no stock for that product (LEFT JOIN).
   * Compare INNER JOIN vs. LEFT JOIN results for products and categories.
3. **Key Concepts**
   * Identify the primary and foreign keys in the order\_items table (Primary Key, Foreign Key).
   * Use foreign keys to retrieve all orders for a specific customer.
4. **Advanced Filtering with Aggregations**
   * Retrieve the top 5 customers with the highest total purchase value (SUM, ORDER BY, TOP).
   * Find all categories where the average price of products is below $100.
5. **Dynamic Data Handling**
   * Replace all NULL descriptions in the products table with "No Description Available" (REPLACE, IS NULL).
   * Use string functions to standardize customer names to uppercase.
6. **Multi-Level Joins**
   * Retrieve store details along with associated staff and their customers (INNER JOIN, LEFT JOIN).
7. **Combining Aggregations and Conditions**
   * Calculate the total revenue for each store and include only those with more than $20,000 in revenue (GROUP BY, HAVING, SUM).
   * Find the highest-priced product in each category.
8. **Mixing Pagination and Filtering**
   * Retrieve products priced between $200 and $500, but show only the 3rd and 4th most expensive products (OFFSET, FETCH, WHERE).
9. **Final Complex Query**
   * Write a single query to display the following:
     + Customer name
     + Total orders they’ve placed
     + Total revenue generated from their orders
     + Include only customers with more than 3 orders and total revenue exceeding $1,000 (JOIN, GROUP BY, HAVING, SUM, COUNT).