**EXERCISE 7**

You are tasked with creating a database table to store information about students. The table should be named **"StudentInfo"** and have the following structure:

* **student\_id (Primary Key):** A unique identifier for each student.
* **first\_name:** The first name of the student.
* **last\_name:** The last name of the student.
* **date\_of\_birth:** The date of birth of the student.
* **gender:** The gender of the student (M/F/O for Male/Female/Other).
* **email:** The email address of the student (unique).
* **phone\_number:** The phone number of the student.
* Create the "StudentInfo" table with the specified columns and constraints.

**Additionally, insert the following 15 different students' information into the table:**

1. Student ID: 101, First Name: John, Last Name: Smith, Date of Birth: 1998-05-15, Gender: M, Email: john.smith@email.com, Phone Number: 123-456-7890
2. Student ID: 102, First Name: Emily, Last Name: Davis, Date of Birth: 1999-03-20, Gender: F, Email: emily.davis@email.com, Phone Number: 987-654-3210
3. Student ID: 103, First Name: Michael, Last Name: Johnson, Date of Birth: 1997-07-10, Gender: M, Email: michael.johnson@email.com, Phone Number: 555-123-4567
4. Student ID: 104, First Name: Sarah, Last Name: Wilson, Date of Birth: 2000-01-05, Gender: F, Email: sarah.wilson@email.com, Phone Number: 789-987-6543
5. Student ID: 105, First Name: David, Last Name: Brown, Date of Birth: 1996-09-30, Gender: M, Email: david.brown@email.com, Phone Number: 111-222-3333
6. Student ID: 106, First Name: Olivia, Last Name: Lee, Date of Birth: 1999-12-18, Gender: F, Email: olivia.lee@email.com, Phone Number: 444-555-6666
7. Student ID: 107, First Name: Ethan, Last Name: Martinez, Date of Birth: 1998-11-25, Gender: M, Email: ethan.martinez@email.com, Phone Number: 777-888-9999
8. Student ID: 108, First Name: Sophia, Last Name: Taylor, Date of Birth: 2001-02-14, Gender: F, Email: sophia.taylor@email.com, Phone Number: 222-333-4444
9. Student ID: 109, First Name: Aiden, Last Name: Miller, Date of Birth: 1997-04-12, Gender: M, Email: aiden.miller@email.com, Phone Number: 666-777-8888
10. Student ID: 110, First Name: Emma, Last Name: Anderson, Date of Birth: 2002-08-07, Gender: F, Email: emma.anderson@email.com, Phone Number: 333-444-5555
11. Student ID: 111, First Name: Benjamin, Last Name: Harris, Date of Birth: 1995-06-22, Gender: M, Email: benjamin.harris@email.com, Phone Number: 888-999-0000
12. Student ID: 112, First Name: Mia, Last Name: Johnson, Date of Birth: 1998-10-01, Gender: F, Email: mia.johnson@email.com, Phone Number: 999-000-1111
13. Student ID: 113, First Name: William, Last Name: White, Date of Birth: 2003-03-12, Gender: M, Email: william.white@email.com, Phone Number: 000-111-2222
14. Student ID: 114, First Name: Ava, Last Name: Robinson, Date of Birth: 1999-04-28, Gender: F, Email: ava.robinson@email.com, Phone Number: 111-222-3333
15. Student ID: 115, First Name: James, Last Name: Turner, Date of Birth: 1996-12-05, Gender: M, Email: james.turner@email.com, Phone Number: 222-333-4444

**ASCII and CHR Functions:**

1. Retrieve the student\_id, first\_name, and last\_name from the StudentInfo table. Use the ASCII function to find the ASCII values of the first characters of both first and last names for each student.
2. You want to find the ASCII value of the '@' symbol in each student's email address. Write an SQL query to retrieve the student\_id, email, and the ASCII value of '@' using the ASCII function.
3. Display the student\_id and first\_name from the StudentInfo table. Use the CHR function to create a new column containing a special character for each student, such as a heart symbol (♥).
4. Calculate the sum of ASCII values for the characters in each student's first name. Retrieve the student\_id, first\_name, and the calculated sum using the ASCII function and aggregation.
5. Retrieve the student\_id, last\_name, and the ASCII value of the last character in the last name for each student using the ASCII function.

**CONCAT Function:**

1. Create a query that retrieves the student\_id, first\_name, and last\_name from the StudentInfo table. Use the CONCAT function to display the full names in the format "Last Name, First Name."
2. You want to create email addresses for students based on their first names. Retrieve the student\_id, first\_name, and a new column with email addresses using the CONCAT function. Assume the email domain is '@example.com'.
3. Display the student\_id, email, and a new email address for each student created by concatenating their student\_id with '@university.com' using the CONCAT function.
4. Retrieve the student\_id, first\_name, and last\_name from the StudentInfo table. Use the CONCAT function to create a new column displaying the first name followed by the last name without a space.
5. You need to generate usernames for students by combining their first names and the last two digits of their student\_id. Retrieve the student\_id, first\_name, and the generated usernames using the CONCAT function.

**LOWER and UPPER Functions:**

1. Display the student\_id and email from the StudentInfo table. Convert the email addresses to lowercase using the LOWER function.
2. Retrieve the student\_id, first\_name, and last\_name from the StudentInfo table. Use the UPPER function to display the full names in uppercase.
3. Calculate the total number of students with lowercase email addresses in the StudentInfo table using the LOWER function and COUNT aggregation.
4. Retrieve the student\_id, email, and first\_name. Convert the email addresses to uppercase and display them alongside the original first names using the UPPER function.
5. You want to display the student\_id, email, and last\_name from the StudentInfo table. Convert the email addresses to uppercase and remove any leading and trailing spaces using the UPPER function and TRIM function.

**COUNT, AVG, MAX, MEDIAN, MIN, and SUM Functions:**

1. Calculate the total count of students in the StudentInfo table.
2. Determine the average age of students based on their date of birth and display it.
3. Find the maximum and minimum lengths of students' email addresses and display these values.
4. Determine the sum of ASCII values of the first character of each student's last name and display the result.

**TRIM Function:**

1. You have a column named description in a table that contains text data. You want to remove any leading and trailing spaces from the values in this column. Write an SQL query using the TRIM function to achieve this.
2. Retrieve the names of all students in the StudentInfo table. Some names have extra spaces at the beginning and end. Write an SQL query using the TRIM function to display the names without leading and trailing spaces.
3. In a table that stores product names, you notice that some names have unnecessary spaces. Write an SQL query using the TRIM function to remove all leading and trailing spaces from the product names.
4. You need to list all cities from a table of addresses. However, some city names have leading spaces. Write an SQL query using the TRIM function to display the city names without any leading spaces.
5. Retrieve a list of email addresses from the StudentInfo table. Some email addresses have extra spaces in them. Use the TRIM function to remove any leading and trailing spaces from the email addresses in your query.

**LTRIM Function:**

1. You have a column called product\_code in a table where some values have extra spaces at the beginning. Write an SQL query using the LTRIM function to remove leading spaces from the product codes.
2. In a table containing book titles, some titles have leading spaces. Write an SQL query using the LTRIM function to display the book titles without any leading spaces.
3. Retrieve a list of employee usernames from a table. Some usernames have leading spaces. Use the LTRIM function to remove these leading spaces in your query.
4. You are working with data from a sensor, and the sensor IDs sometimes have extra spaces at the beginning. Write an SQL query using the LTRIM function to remove any leading spaces from the sensor IDs.
5. In a table that stores product names, some names have leading spaces that need to be removed. Write an SQL query using the LTRIM function to clean the product names.

**RTRIM Function:**

1. You are dealing with a table that contains URLs, and some of them have trailing spaces. Write an SQL query using the RTRIM function to remove any trailing spaces from the URLs.
2. In a table that stores file paths, some paths have trailing spaces that need to be eliminated. Write an SQL query using the RTRIM function to display the file paths without trailing spaces.
3. Retrieve a list of employee email addresses from a table. Some email addresses have trailing spaces. Use the RTRIM function to remove these trailing spaces in your query.
4. You have a table with customer names, and some names have trailing spaces. Write an SQL query using the RTRIM function to display the customer names without any trailing spaces.
5. In a table containing postal codes, some codes have trailing spaces that need to be cleaned. Write an SQL query using the RTRIM function to remove the trailing spaces from the postal codes.

**TRIM Function:**

1. You are dealing with a table that contains employee usernames. Some usernames have both leading and trailing spaces. Write an SQL query using the TRIM function to retrieve the employee\_id and cleaned usernames for all employees.
2. In a table storing company names, you notice that some names have both leading and trailing spaces. Write an SQL query using the TRIM function to display the unique company names without leading and trailing spaces.
3. Retrieve a list of product codes from the StudentInfo table. Some product codes have both leading and trailing spaces. Use the TRIM function to clean the product codes in your query.
4. In a table containing postal codes, some codes have both leading and trailing spaces that need to be removed. Write an SQL query using the TRIM function to clean the postal codes.
5. You have a table with customer names, and some names have both leading and trailing spaces. Write an SQL query using the TRIM function to display the customer names without any leading and trailing spaces.

**SUBSTR Function:**

1. Retrieve the first three characters of each student's first name from the StudentInfo table using the SUBSTR function. Display the student\_id and the extracted substrings.
2. You need to extract the last two characters from each student's last name. Write an SQL query using the SUBSTR function to retrieve the student\_id, last\_name, and the extracted substrings.
3. Display the student\_id, email, and only the domain part of each student's email address using the SUBSTR function. Assume that the domain is the characters after the '@' symbol.
4. Calculate the length of the first five characters in each student's email address. Retrieve the student\_id, email, and the length of the substrings using the SUBSTR and LENGTH functions.
5. Retrieve the student\_id, first\_name, and the third to fifth characters of each student's first name using the SUBSTR function.

**NVL Function:**

1. In the StudentInfo table, some students have missing values for their phone numbers (NULL). Use the NVL function to display 'Not Available' for students with no phone number. Retrieve the student\_id, first\_name, and phone numbers.
2. Calculate the age of each student based on their date of birth, and for students with missing birthdates (NULL), display 'Age Unknown' using the NVL function. Retrieve the student\_id, first\_name, and the calculated age.
3. You want to categorize students as 'Male' or 'Female' based on their gender, but some have NULL values. Use the NVL function to categorize students with NULL gender values as 'Unknown'. Retrieve the student\_id, first\_name, and the categorized gender.
4. Display the student\_id, email, and for students with NULL email addresses, show 'No Email' using the NVL function.
5. You have a table that stores product prices, and some prices are missing (NULL). Use the NVL function to display 'Price Not Available' for products with no price. Retrieve the product names and prices.

**NVL2 Function:**

1. You want to calculate the bonus for employees. If an employee's performance is 'Excellent,' the bonus is 1000; otherwise, it's 0. Use the NVL2 function to calculate the bonus. Retrieve the employee\_id, first\_name, performance, and the calculated bonus.
2. In a table that stores order quantities, some quantities are missing (NULL). Use the NVL2 function to calculate the adjusted quantity. If the quantity is missing, assume it's 0. Retrieve the order IDs and adjusted quantities.
3. You need to categorize students based on their age. If a student is 18 or older, categorize them as 'Adult'; otherwise, categorize them as 'Minor.' Use the NVL2 function to categorize students with NULL birthdates as 'Unknown.' Retrieve the student\_id, first\_name, and the categorized age.
4. Calculate the total score for students. If a student's score is missing (NULL), use the NVL2 function to assume their score is 0. Retrieve the student\_id, first\_name, and the total score.
5. In a table storing product quantities, some quantities are missing (NULL). Use the NVL2 function to calculate the adjusted quantity. If the quantity is missing, assume it's 1. Retrieve the product names and adjusted quantities.

**LENGTH Function:**

1. Calculate the length of each student's first name in the StudentInfo table. Retrieve the student\_id, first\_name, and the length of the first names.
2. You have a table that stores email addresses, and you want to find the length of each email address. Retrieve the email addresses and their lengths using the LENGTH function.
3. Determine the number of characters in each student's last name in the StudentInfo table. Retrieve the student\_id, last\_name, and the length of the last names.
4. Calculate the length of each city name in a table that stores cities. Retrieve the city names and their lengths using the LENGTH function.
5. You want to find the length of each product name in a table. Retrieve the product names and their lengths using the LENGTH function.

**SOUNDEX Function (StudentInfo Table):**

1. You have a requirement to find students in the StudentInfo table who may have similar-sounding last names. Write an SQL query using the Soundex function to display the student\_id, last\_name, and Soundex codes for students with last names that sound alike.
2. Use the Soundex function in the StudentInfo table to calculate the Soundex codes for each student's last name. Retrieve the student\_id, last\_name, and Soundex codes.
3. In the StudentInfo table, some students may have last names that sound similar but are spelled differently. Write an SQL query with the Soundex function to identify such students. Display the student\_id, last\_name, and Soundex codes.
4. You are tasked with finding potential duplicate student records in the StudentInfo table based on similar-sounding last names. Write an SQL query using the Soundex function to retrieve the student\_id, last\_name, and Soundex codes for students with last names that sound alike.