**SET OPERATIONS:**

**QUERY:**

*SELECT episode, title FROM `elements-by-episode` WHERE title LIKE '%WINTER%'*

*UNION ALL*

*SELECT episode, title FROM `elements-by-episode`s WHERE title LIKE '%SUMMER%'*

*ORDER BY title;*

**EXPLANATION:**

The UNION ALL set operation command was used to combine the results of 2 SELECT statements. UNION ALL the titles that have the words “winter” and “summer”.

**SCREENSHOT**:

Graphical user interface, text, application

Description automatically generated

**SUBQUERIES:**

**QUERY:**

*SELECT \* FROM `elements-by-episode` WHERE title IN*

*(SELECT title FROM `elements-by-episode` WHERE AURORA\_BOREALIS=1);*

**EXPLANATION:**

Subqueries are queries within parentheses. Within the parentheses the SELECT commands with titles WHERE the column Aurora\_Borealis = 1. This query returned 2 row that has = 1 in the Aurora Borealis column.

**SCREENSHOT:Graphical user interface, application, Word

Description automatically generated**

**ORDER OF OPERATION OF QUERIES:**

**QUERY:**

*SELECT EPISODE, TITLE, TREES FROM `elements-by-episode` WHERE SUN >= 1*

*ORDER BY TITLE DESC;*

**EXPLANATION**:

The following SELECTED columns WHERE SUN equal to or greater than 1. The order of operation query ORDER BY command was used to order the title names by descending order.

**SCREENSHOT:Graphical user interface, text, application

Description automatically generated**

**CREATING, ALTERING, AND DROPPING TABLES:**

**QUERY:**

*CREATE TABLE ARTIST\_ELEMENTS LIKE `elements-by-episode`;*

*SELECT \* FROM ARTIST\_ELEMENTS;*

*ALTER TABLE ARTIST\_ELEMENTS ADD Color varchar(25) ;*

*DROP TABLE artist\_elements;*

**EXPLANATION:**

Created a table name ARTIST\_ELEMENTS that is just like the `elements-by-episode table.

After selecting the new table, the table was altered by adding a new column named Color with a varchar of 25. After successfully altering the table; the table was deleted with the DROP TABLE command.

**SCREENSHOT:** Graphical user interface, text, application

Description automatically generated

**ASSOCIATIONS:**

**QUERY:**

*CREATE TABLE IF NOT EXISTS WATER*

*AS SELECT EPISODE, TITLE, LAKE, LAKES, OCEAN, RIVER, WATERFALL FROM elements;*

*SELECT elements.EPISODE, water.TITLE*

*FROM elements*

*INNER JOIN water ON elements.EPISODE = water.EPISODE;*

*JOINS AND MULTIPLE TABLE JOINS:*

**EXPLANATION**:

An association defines a relationship between two [entity objects](https://docs.oracle.com/cd/A97335_02/apps.102/bc4j/developing_bc_projects/Glossary.htm#entity_object) based on common [attributes](https://docs.oracle.com/cd/A97335_02/apps.102/bc4j/developing_bc_projects/Glossary.htm#attribute). The relationship can be one-to-one or one-to-many; you can use two one-to-many associations to implement a many-to-many relationship. A new table named WATER was created using the INNER JOIN command. This query shows all the titles from the WATER table that have an association through the water.episode column with the elements.episode column.

**SCREENSHOT:Graphical user interface, text, application, email

Description automatically generated**

**JOINS / MULTIPLE TABLE JOINS:**

**QUERY:**

*SELECT elements.EPISODE, elements.TITLE, water.TITLE, water.Lake, water.OCEAN, water.RIVER, water.WATERFALL*

*FROM elements*

*LEFT JOIN water*

*ON elements.title = water.title*

*ORDER BY WATER.title ASC;*

**EXPLANATION:**

The LEFT JOIN keyword returns all records from the left table (elements table), and the matching records from the right table (water table).

**SCREENSHOT:Application

Description automatically generated with medium confidence**