**Set operations:**

The Process Set Operations are a set of operations that allow for the manipulation of data in the process space. The process set operations allow you to create, delete, modify, and list your processes. You can also change the name of a process or its configuration settings. These operations can be performed on all processes in your system at once or one-by-one with just one click. These operations include:

-Add: adds a new process to an existing set

-Delete: deletes a process from an existing set

-Find: finds a process by its PID or name

-FindbyName: finds a process by its name

-List: lists all processes in the current set

**Subqueries:**

Subqueries are a way to filter data. They are used to create more complex queries that run over the data and create new results. For example, if you have a list of some products and want to find out how many of them sold in each state, you can use a subquery to get all the products that sold in each state and then combine them into one result set. A process subquery is a query that helps you create a process-based SQL statement. This statement allows you to retrieve data from multiple tables, where each table has an associated process that describes how the data is processed in a step-by-step manner.

A process subquery is used for two main reasons: first, to create an aggregate function or join based on the data in one or more tables; and second, to execute a step-by-step process through one or more databases.

The syntax for creating a process subquery is as follows:

SELECT DISTINCT [ALL] FROM <table1> JOIN <table2> ON <column1> = <column2> WHERE <condition1> OR <condition2>. It is important to note that the syntax shown here will always return only one row of data per query execution.

**Order of operation of queries:**

A general overview of the process order of operation of queries

The process order of query execution is as follows:

1. The request is sent to the database server and processed.

2. If the database server cannot accept or reject the request, it returns a message to the client application with an error code that describes why it was unable to accept or reject the request.

3. If the database server accepts a request, it calls a method in an object called Query Handler on its own memory space and passes it the request data as parameters. Then, it calls a method in an object called Query Processor on its own memory space and passes it an array that contains all the parameters from step 2, along with any other data that may be needed by this query, such as user-supplied values for fields not found in database tables.

4. The Query Processor then calls another method in an object called QuerySqlCodeGenerator on its own memory space and passes it some parameters that depend on which SQL dialect is being used (e.g., for PostgreSQL). This method then generates all the SQL code necessary to execute

**Creating, altering, and dropping tables:**

Creating a Table

To create a table, you must first have a database. You can create a new database by using the CREATE DATABASE statement. Once your database has been created, you can create tables within it by using the CREATE TABLE statement.

CREATE TABLE `bobross`.`second\_db` ( `id` INT NOT NULL , `firstname` VARCHAR(50) NOT NULL , `lastname` VARCHAR(50) NOT NULL , `age` INT NOT NULL , PRIMARY KEY (`id`));

Altering Tables

To alter an existing table, you must first have permissions to do so. You can grant yourself these permissions by using the GRANT command with appropriate arguments. Alternatively, if the table belongs to another user account, that user must grant those permissions before altering it will succeed.

Graphical user interface, application

Description automatically generated

Dropping Tables

To drop an existing table from your database, you must use the DROP TABLE statement with appropriate arguments and parameters.

**Associations:**

The process association is a connection between two processes. This connection can be either direct or indirect. A direct process association is a connection between two processes that are directly connected by one channel. An indirect process association is a connection between two processes that are indirectly connected by one or more channels.

**Joins and multiple table joins:**

Join operations are used to combine the rows of two or more tables.

In SQL, if you want to join two tables, you must have a common column between the two tables. A column is the data in the table that describes a single record. If the tables have no common columns, they cannot be joined together. When you join multiple tables together, it is called a multiple table join. When using a multiple table join, there are several types of joins that can be performed: inner joins, outer joins, semi-joins, and anti-joins. These are explained below.

An inner join returns only those rows for which there is an exact match between the two tables being joined; all other rows are omitted from the result set (ignored). If any of the matching rows in one table do not have matching rows in another table, then these missing rows are also omitted from the result set (ignored). An inner join can be thought of as an associative operation because it uses both its left and right operands to form a single result set (or association).

An outer join returns all rows from both its left and right operands.