Certainly! Here are some basic SQL syntax examples that can be used in PostgreSQL, a popular open-source relational database:

1. Creating a Table:

To create a new table in a PostgreSQL database:

```sql
CREATE TABLE employees (
    id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100) UNIQUE,
    age INT
);
```

This SQL statement creates a table named "employees" with columns for ID, first name, last name, email, and age. It designates the "id" column as a primary key, using the SERIAL data type for auto-incrementing.

2. Inserting Data:

To insert data into the "employees" table:

```sql
INSERT INTO employees (first_name, last_name, email, age)
VALUES ('John', 'Doe', 'john@example.com', 30);
```

This SQL statement inserts a new record into the "employees" table with specific values for the columns specified.

3. Selecting Data:

To retrieve data from the "employees" table:

```sql
SELECT * FROM employees;
```

This SQL statement fetches all columns and rows from the "employees" table.

4. Filtering Data:

To retrieve specific data based on conditions:

```sql
SELECT * FROM employees WHERE age > 25;
```

This SQL statement retrieves all columns and rows from the "employees" table where the "age" is greater than 25.

5. Updating Data:

To update existing data in the "employees" table:

```
UPDATE employees SET age = 32 WHERE first_name = 'John' AND last_name = 'Doe';
```

This SQL statement updates the "age" column for the employee named 'John Doe'.

6. Deleting Data:

To delete data from the "employees" table:

```
DELETE FROM employees WHERE age < 25;
```

This SQL statement deletes rows from the "employees" table where the "age" is less than 25.

These basic SQL commands demonstrate how to create tables, insert data, retrieve data, update records, and delete records in a PostgreSQL database. The syntax is fundamental and can be extended with more complex operations and clauses as needed.