Hack postgres Source Code: Vol I

Chapter 1 : C & Rust

1.1 Strings in postgres

C:

```c
#include<stdio.h>

int
main() {

    /*
    * Strings are not primitive types in C.
    * These are character arrays that ends with null character \0
    * Size of string is defined considering null character.
    * Strings are initialized in different types.
    * Characters in string can be retrieved using index.
    * Strings use format specifies %s.
    */

    /*
    * Array type initialization
    */
    char _string_t1[4] = {'s', 't', 'r', '\0'};
    char _string_t2[]  = {'s', 't', 'r', '\0'};

    /*
    * String type initialization with quotes
    * In this type, null character is automatically added by C internally
    */
    char _string_t3[4] = "str";
    char _string_t4[] = "str";

    printf ("%s\n", _string_t1);
    printf("%c\n", _string_t1[0]); // character retreival from string


    return 0;

}
```

Rust:

```rust
fn main() {


    /*
    * Strings in rust categorized as string literals
    * and string objects.
    * String literals are static in nature.
    * String Objects are mutable.
    * Strings don't follow index notation to retrieve character
    * from string.
    */

    // String literals type &str or &'static str
    let _string_literal: &'static str = "str";
    println!("{}", _string_literal);


    // String object
    let _string_object_t1 = String::new(); // creates empty string
    let _string_boject_t2 = String.from("str");  // from string literal

    println!("{} and {}", _string_object_t1, _string_boject_t2);


}
```