

# Hack postgres Source Code: Vol I

---

## Chapter 1 : C & Rust

---

### 1.9 Structs in postgres

C:

```
#include<stdio.h>

/*
 * Structs are custom data types created out of primitive types in
 * most of the cases.
 * Struct members are accessed with `.` operator
 * Structs are generally intialized out of the main function.
 * Structs follow camelcase convention for their names.
 */

struct Employee {
    int emp_id;
    char emp_name[64];
};

int
main() {

    struct Employee emp_1 = {101, "John"};

    printf("%d and %s\n", emp_1.emp_id, emp_1.emp_name);

    return 0;

}
```

Rust:

```
struct Employee {
    emp_id: i32,
    emp_name: String,
}

fn main() {

    //Creating immutable struct
    let emp_1 = Employee { emp_id: 101, emp_name: String::from("John")};
```

```
println!("{}", emp_1.emp_id, emp_1.emp_name);

//Creating mutable struct
let mut emp_2 = Employee {emp_id:0, emp_name:String::from("Dummy")};

emp_2.emp_id = 102;
emp_2.emp_name = String::from("Jane");

println!("{}", emp_2.emp_id, emp_2.emp_name);

}
```

## Observations;

1. C and Rust follow same path while dealing with the function at core level.
2. There is only slight difference between C & Rust's syntax.