A union expression does not have exactly one field.

Erroneous code example:

```
union Bird {
    pigeon: u8,
    turtledove: u16,
}

let bird = Bird {}; // error
let bird = Bird { pigeon: 0, turtledove: 1 }; // error
```

The key property of unions is that all fields of a union share common storage. As a result, writes to one field of a union can overwrite its other fields, and size of a union is determined by the size of its largest field.

You can find more information about the union types in the Rust reference.

Working example:

```
union Bird {
    pigeon: u8,
    turtledove: u16,
}

let bird = Bird { pigeon: 0 }; // OK
```