

Students.	Teachers	Marks
Id,	id	id
name,	name	marks
		subName
		stdId
		tcherID

Select Sum(marks) from Marks where stdId in (select id from students ss where ss.name="Jagadeesh") ;

Select name from teacher t inner join marks mm where mm.subName = "maths";

Select name,subName,marks from marks m join students s on m.stdId = s.id where m.marks=(select MAX(marks) from marks mmm groupBy mmm.subName having m.subname = mmm.subname);

formName—inputform
function validation(){

```

var name = document.inputform.name.value;
var password = document.inputform.password.value;
var phone = document.inputform.phone.value;
var age = document.inputform.age.value;
var email = document.inputform.email.value;

```

```

var ap = email.indexOf("@");
var dp = email.lastIndexOf(".");

```

```

if(name==null || name==""){
    alert("type a name to proceed");
    return false;
}else if(password.length<10){
    alert("password must be of 10 or more chars");
    return false;
}else if(isNaN(phone)){
    alert("chars are not allowed for phone numbers");
    return false;
}else if(age==null || age=="" || age<18){
    alert("age. Criteria fails...please enter a valid age");
    return false;
}else if(ap<1 || dp+2>=email.length || dp<ap+2){
    alert("invalid email");
    return false;
}
}

```

```

@Table
@Entity(name="students")
Public class student extends StudentDB{

```

```
@NotNull
@Column(name="StudentName",nullable=false)
protected String stName;

@Column(name="email",nullable=false, length=120)
Protected String email;

@Column(name="password",nullable=false, length=10)
Protected String password;

}
```

```
main(){

FileReader r = new FileReader("info.properties");
Properties props = new Properties();
Props.load(r);
String result1 = props.getProperty("Key1");

}

Set a = props.entrySet()
```