```
class InsurancePlan {
    int planNo;
    String planName;
    int tenure;
    double premiumAmount;
    double insurredAmount;
    String status;
    // accessors
}
```

```
class MedicalInsurancePlan
extends InsurancePlan {
   int copay;
   String coverageType;
   // accessors
}
```

```
class AccidentalInsurancePlan
extends InsurancePlan {
    int disabilityCoveragePercentage;
    boolean internationalCoverage;
}
```

```
insurance_plan
plan_no (pk)
plan_nm
tenure
premium_amt
insurred_amt
status
```

```
medical_insurance_plan
medical_insurance_plan_no (pk) (fk)
copay
coverage_type
```

```
accidental_insurance_plan
accidental_insurance_plan_no (pk) (fk)
disability_coverage_percentage
international_coverage
```

Table Strategy:
In Table per Subclass, for the super class, create one table representing the attributes of the class as columns within the table and per each subclass within the hierarchy create another table inrelationship with superclass table.

Let us examine how to perform persistence operations for the above entity classes onto those tables we designed:

#1. SAVE/PERSIST:
1.1 Superclass
How to persist the data of an Superclass object?
The superclass object should be persisted into superclass represented table directly. Here InsurancePlan object should be stored in insurance_plan table directly

1.2 Subclass
How to persist the data of an Subclass object?
The subclass object contains inherited fields of superclass as well, so to persist all of the data together we need to insert #2 records in superclass represented table and subclass
Store the inherited fields that are part of the subclass into Superclass represented table, then pick the pk of the record and store it as pk,fk in the subclass represented table a

for eg.. to persist MedicalInsurancePlan object, pick the superclass attributes and insert a record in insurance_plan table. then pick the pk with which the record has been inserted

insurance_plan

plan_no	plan_nm	tenure	premium_amt	insurred_amt	status
1	ip1	12	2000	200000	A
2	mip1	24	4000	400000	A
3	aip1	36	6500	800000	A

medical_insurance_plan

medical_insurance_plan_no	copay	coverage_type
2	10	in-patient

accidental_insurance_plan

accidental_insurance_plan_no	disability_coverage_percentage	international_coverage
3	50	0

#2. GET/FETCH:
2.1 Subclass:
How to fetch/get the data of a subclass object?
inorder to query the data for a subclass, the record should be existing in both superclass table and its corresponding subclass table as well because while persisting the data for
a subclass we are inserting the #2 records into superclass table/and corresponding subclass table.
So we need to query the data for a subclass by joining superclass table with corresponding subclass table

MedicalInsurancePlan mip = session.get(MedicalInsurancePlan.class, 2);

select ip.*,mip.* from insurance_plan ip inner join medical_insurance_plan mip on ip.plan_no = mip.medical_insurance_plan_no where ip.plan_no = 2;

MedicalInsurancePlan mip = session.get(MedicalInsurancePlan.class, 3);
we cannot express one subclass object interms of another subclass, that is not supported by polymorphic access. so when we query the planNo = #3, that belongs to
AccidentalInsurancePlan interms of MedicalInsurancePlan above, it will not let us access the AIP record interms of MIP object because we are using inner join in fetching the data
for MedicalInsurancePlan
So unless the record exists in insurance_plan table & medical_insurance_plan table it will return the record.

2.2 Superclass:
How to fetch the data for a Superclass?
In general we can say a record belongs to the super class, when it exists only in Superclass represented table.

InsurancePlan ip = session.get(InsurancePlan.class, 1); // this should fetch the record from insurance_plan table and return InsurancePlan object since #1 belongs to InsurancePlan

But what about a Subclass object can be accessed interms of Superclass, is this should be supported or not?
Yes, this is polymorphic retrieval and should be supported when the classes are in Inheritance relationship

InsurancePlan ip = session.get(InsurancePlan.class, 2);
asper the above statement, we said when we are accessing the data for a superclass object, we should query only in superclass table. In such case when we query for planNo = 2, if
it looks only in Insurance_plan table, how does it knows that #2 belongs to MedicalInsurancePlan object and returns it?

Then how to retrieve the data for a Superclass reference Type, that supports even polymorphic access?