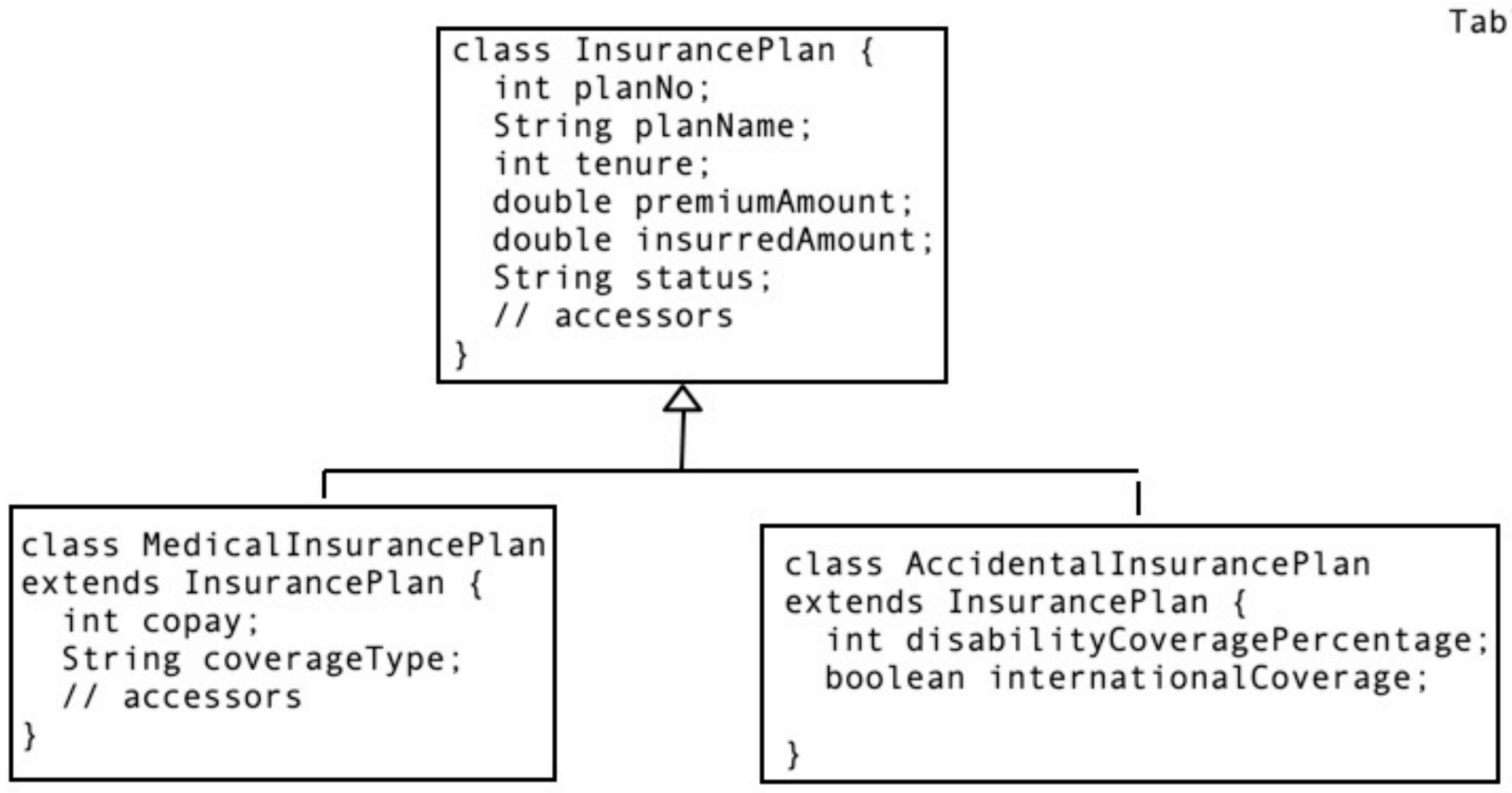
Table per concrete class



```
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
plan_nm
tenure
premium_amt
insurred_amt
status
```

```
medical_insurance_plan
plan_no
plan_nm
tenure
premium_amt
insurred_amt
status
copay
coverage_type
```

```
accidental_insurance_plan
plan_no
plan_nm
tenure
premium_amt
insurred_amt
status
disability_coverage_percentage
international_coverage
```

Tables with data:
----

insurance_plan

plan_no	plan_nm	tenure	premium_amt	insurred_amt	status
1	ip1	12	2000	200000	A

medical_insurance_plan

plan_no	plan_nm	tenure	premium_amt	insurred_amt	status	copay	coverage_type
2	mip1	24	3000	250000	A	10	in_patient

accidental_insurance_plan

plan_no	plan_nm	tenure	premium_amt	insurred_amt	status	disability_coverage_percentage	international_coverage
3	aip1	36	5000	800000	A	50	0

How does the entity classes with inheritance relationship are modeled in tables?
Strategy:
Per each concrete class in the hierarchy there is an seperate table created. For superclass, a table with superclass attributes and per each subclass a table with superclass attributes + subclass attributes being created.
Incase if the superclass is an abstract class, we dont need a table for holding the superclass object data because if it is abstract class we cannot create the objects for the Superclass, so we dont need table for holding those objects of data. Anyhow for each subclass we have an seperate table for holding their data.

How to perform persistence operations for these entity classes of the hierarchy based on the Mapping strategy: Table per concreate class?
1. PERSIST/SAVE:
  1.1 Superclass: How to store the data of a super class?
  directly store the data in the superclass corresponding table
  1.2 Subclass: How to store the data of a Subclass?
  directly store the data into the corresponding table of that subclass
To support polymorphic retrieval of the data, while inserting the data for any of the classes in the hierarchy, we should ensure the id should be unique across the tables of the hierarchy. To do that we should not use identity generator.
Note: dont use identity generator

we can use increment, sequence, table generator
increment = it queries the max(id) from all the tables based on UNION query and increments by 1 and returns,
sequence = we should use same sequence for all the tables in the hierarchy
table generator = the pkColumnName to be used for generating the id for all the classes in the hierarchy should be same.

2. FETCH/GET:
  1.1 Subclass: How to query the data for an given Subclass?
  MedicalInsurancePlan mip = session.get(MedicalInsurancePlan.class, 1);
  while querying the data for a subclass, we can directly fetch the data from the corresponding subclass table, since each Entity class has its own respective table into which the data is being persisted.

  1.2 Superclass: How to query the data for a Superclass?
  InsurancePlan ip = session.get(InsurancePlan.class, 1);
  Looks like when we are querying the data for a superclass type we can directly goto superclass table for fetching the data, since everyclass in the hierarchy has its own table.
But in inheritance relationship, we should support polymorphic retrieval, means a subclass object should be accessed interms of superclass reference Type. Since all the entity classes in the hierarchy are stored separately, they have their own independent identity which could be duplicated across the hierarchy

so when we query the entity object with an id based on superclass type, we cannot determine the object belongs to which subclass, since it is found in all the three tables. So looks polymorphic access is not supported.
Is there a way we can accomplish polymorphic retrieval?

when we are querying the data for a super ClassType, if we go and look for the data in superclass table only, then we would not be able to support polymorphic access. To support accessing any of the subclass objects via the super classType, we need to query the data from all the tables in the hierarchy and see the record exists in which table for the given id and return appropriate object. But the tricky thing is since we have separate tables for every class in the hierarchy the pk value will be repeated/same across the classTypes, so we cannot uniquely identify a record of given id belongs to which entity classType, since we may find the records in all the tables of that id.

So to support polymorphic retrieval of the data based on Super ClassType, we need to generate the id while inserting the data for any of the classes in the hierarchy uniquely.

So while accessing the data for the super classType, query the data from all the three tables of the hierarchy using union query and find the record exists with that id in which table based on that return corresponding entity object

```
InsurancePlan ip = session.get(InsurancePlan.class, 2);
select ip.*, null,null,null,null,0 from insurance_plan ip where ip.plan_no = 2
union
select mip.*, null, null, 1 from medical_insurance_plan mip where mip.plan_no = 2
union
select aip.*, null, null, 2 from accidental_insurance_plan aip where aip.plan_no = 2
```

2 mip1 24 3000. 300000 10 in_patient null null [1] = so by looking at the [1] as classType it identifies the record as medical_insurance_plan type and creates MedicalInsurancePlan object and returns it.