



persist:

parent/super class Payment:
parent class attributes into corresponding columns, leaving child class attributes with null values. Along with that store discriminator-column value as "cash".

sub-class (CardPayment/ChequePayment):
parent class attributes + corresponding child-class attributes (leaving other subclass attributes as NULL) with discriminator-column value specific to the child class object we are storing
CardPayment = card
ChequePayment = cheque

query/fetch:

Sub-class (CardPayment/ChequePayment):

To ensure the object we are querying is pertaining to that subclass only we need to consider discriminator-value of that sub-class we are querying in where clause along with primary key value to fetch the record of data for that subclass

select * from payment where id=? and payment_type=(discriminator-value of that sub-class)

Parent class:
as any subclass objects can be queried as super class reference type, to support such polymorphic retrieval the hibernate/jpa only queries the record based on primary key only.

upon fetching the record, it checks the discriminator-column value, based on the value it populates and returns an corresponding class object of the hierarchy.