WE CAN CREATE NUMBER OF PARTITIONS WE WANT
BY LOADING THE DATA INTO THE SAME FILE IT CREATES
NEW FILE IN ITS SUB DIRECTORY WHICH IS POINTING TO HDFS
##
we need to create a new AVSC file by copying same properties of file
in local system
cp orders.avsc sandeep.avsc
---
view sandeep.avsc
Now we need to modifie this avsc file ti improve query performance
1.we need to add a partition column

```
{
  "type" : "record",
  "name" : "CHANGE NAME",
  "doc" : "HIVE AVRO PARTITINED TABLE CHANGE NAME",
  "fields" : [ {
    "name" : "order_id",
    "type" : [ "null", "int" ],  REMOVE COMMA HERE
    "default" : null,
    "columnName" : "order_id",                    DELETE THAT IN STARS
    "sqlType" : "4"  ReMoVe UpTo HeRe       FROM COMA TO SQL TYPE
  }, {
    "name" : "order_date",
    "type" : [ "null", "long" ],
    "default" : null,
    "columnName" : "order_date",
    "sqlType" : "93"
  }, {
    "name" : "order_customer_id",
    "type" : [ "null", "int" ],
    "default" : null,
    "columnName" : "order_customer_id",
    "sqlType" : "4"
  }, {
    "name" : "order_status",
    "type" : [ "null", "string" ],
    "default" : null,
```

```
   "columnName" : "order_status",
   "sqlType" : "12"
  } ],
  "tableName" : "CHANGE NAME"
}
```

==========

Now before creating the table make sure that the table which you are going to take refrence
is existing in the same DATABASE which you are going to create TABLE
as you can see in our DATABASE we have orders and thats why we are taking refrence of orders
while we are LOADING the DATA

==========

Now create table name should be same as mentioned in .AVSC file

==========

```
CREATE TABLE record_sandeep (
order_id int,
order_date bigint,
order_customer_id int,
order_status string
)
PARTITIONED BY (order_month string)
STORED AS AVRO
LOCATION 'hdfs:///user/hive/warehouse/retail_stage.db/orders_part_avro'
TBLPROPERTIES
('avro.schema.url'='hdfs://quickstart.cloudera/user/cloudera/avscData/sandeep.avsc'
);
```

==========

DYNAMIC PARTITION

==========

```
set hive.exec.dynamic.partition.mode=nonstrict;
insert into table record_sandeep partition (order_month)
select order_id, order_date, order_customer_id,
order_status,order_month("sandeep") from
orders;
select order_id, order_date, order_customer_id, order_status,
```

substr(from_unixtime(cast(substr(order_date, 1, 10) as int)), 1 ,7) order_month from orders;

HERE 1,7 means jan to july

==========

--validate

==========

=

--validate

dfs -ls /user/hive/warehouse/retail_stage.db/orders_part_avro/*

dfs -ls /user/hive/warehouse/retail_stage.db/orders_part_avro/