Bucketing Table

- Create bucketed table -
    - **create table cases_bucket(Case_Id int, Hospital_Code int, Patient_Id int, Ward_Type char(1), Ward_Facility_Code char(1), Bed_Grade int, Type_of_Admission string, Visitors_with_Patient int, Age_Group string, Admission_Deposit float, Stay string)**

      partitioned by (Department string, Severity_of_Illness string)

      clustered by (patient_id) into 10 buckets

      row format delimited

      fields terminated by '$'

      TBLPROPERTIES ("serialization.null.format"="");
    - We are creating 10 buckets based on the patient_id within each partition.
- Put file to HDFS -
    - **hdfs dfs -put /home/itv180149/Hive/Partitions/Hospital_Data/case.txt /user/itv180149**
- Load data into Hive table -
    - **load data inpath "/user/itv180149/case.txt" into table cases_bucket;**
    - Notice that since we are bucketing the data into 10 buckets, Hive is using 10 reducers to accomplish this job.
- Let's check the table description -
    - **describe formatted cases_bucket;**
    - Notice the partitions and buckets here.

- We have 10 buckets or files within each partition -
    - **hdfs dfs -ls /user/itv180149/warehouse/patient_db.db/cases_bucket/department=anesthesia/severity_of_illness=Extreme**
- So if we run queries on the partitioned and bucketed columns, then those queries will run much faster.
    - Check the records where the patient id is 306180 and department is gynecology and severity of illness is Minor -
        - **select hospital_code, department, severity_of_illness from cases_bucket where patient_id=306180 and department='gynecology' and severity_of_illness='Minor';**