create table temp\_cricket (col\_value string)

LOAD DATA INPATH '/user/cloudera/ContinousDataset.csv' OVERWRITE INTO TABLE temp\_cricket;

CREATE TABLE cricket (odiID INT,scorecard STRING, team1 STRING, team2 STRING, margin STRING, ground STRING, matchdate STRING,matchyear STRING,winner STRING, host\_country STRING, venue\_t1 STRING, venue\_t2 STRING, innings\_t1 STRING, innings\_t2 STRING)

--drop TABLE cricket;

--drop TABLE temp\_cricket;

select \* from temp\_cricket limit 100

select \* from cricket

insert overwrite table cricket

SELECT

regexp\_extract(col\_value, '^(?:([^,]\*),?){1}',1) odiID,

regexp\_extract(col\_value, '^(?:([^,]\*),?){2}',1) scorecard,

regexp\_extract(col\_value, '^(?:([^,]\*),?){3}',1) team1,

regexp\_extract(col\_value, '^(?:([^,]\*),?){4}',1) team2,

regexp\_extract(col\_value, '^(?:([^,]\*),?){5}',1) margin,

regexp\_extract(col\_value, '^(?:([^,]\*),?){6}',1) ground,

regexp\_extract(col\_value, '^(?:([^,]\*),?){7}"',1) matchdate,

regexp\_extract(col\_value, '^(?:([^,]\*),?){8}"',1) matchyear,

regexp\_extract(col\_value, '^(?:([^,]\*),?){9}',1) winner,

regexp\_extract(col\_value, '^(?:([^,]\*),?){10}',1) host\_country,

regexp\_extract(col\_value, '^(?:([^,]\*),?){11}',1) venue\_t1,

regexp\_extract(col\_value, '^(?:([^,]\*),?){12}',1) venue\_t2,

regexp\_extract(col\_value, '^(?:([^,]\*),?){13}',1) innings\_t1,

regexp\_extract(col\_value, '^(?:([^,]\*),?){14}',1) innings\_t2

FROM temp\_cricket

--INSERT OVERWRITE TABLE cricket SELECT \* FROM cricket WHERE 1=0;

--DROP TOP (1) FROM cricket;

ALTER TABLE cricket SET TBLPROPERTIES ("skip.header.line.count"="1");

select count(\*) from cricket;

SELECT count (DISTINCT scorecard) FROM cricket;

SELECT count(DISTINCT team1) as teams FROM cricket;

SELECT count (DISTINCT team2) FROM cricket;

SELECT winner,count(\*) as num\_wins from cricket GROUP BY winner ORDER BY num\_wins DESC LIMIT 1;

SELECT MAX(margin),winner,count(winner) as CNT from cricket GROUP BY winner ORDER BY cnt desc limit 1;

SELECT \* FROM (SELECT winner, count(winner) as matches\_won, '1971-1980' as decade FROM cricket where matchyear>=1971 and matchyear<=1980 group by winner order by matches\_won desc limit 1

union all SELECT winner, count(winner) as matches\_won, '1981-1990' as decade FROM cricket where matchyear>=1981 and matchyear<=1990 group by winner order by matches\_won desc limit 1

union all SELECT winner, count(winner) as matches\_won, '1991-2000' as decade FROM cricket where matchyear>=1991 and matchyear<=2000 group by winner order by matches\_won desc limit 1

union all SELECT winner, count(winner) as matches\_won, '2001-2010' as decade FROM cricket where matchyear>=2001 and matchyear<=2010 group by winner order by matches\_won desc limit 1

union all SELECT winner, count(winner) as matches\_won, '2011-2018' as decade FROM cricket where matchyear>=2011 and matchyear<=2018 group by winner order by matches\_won desc limit 1)

AS cricket ORDER BY cricket.decade ASC

SELECT \* FROM (SELECT winner, count(winner) as matches\_won, '1971-1980' as decade FROM cricket where winner!=Host\_Country and matchyear>=1971 and matchyear<=1980 group by winner order by matches\_won desc limit 1

union all SELECT winner, count(winner) as matches\_won, '1981-1990' as decade FROM cricket where winner!=Host\_Country and matchyear>=1981 and matchyear<=1990 group by winner order by matches\_won desc limit 1

union all SELECT winner, count(winner) as matches\_won, '1991-2000' as decade FROM cricket where winner!=Host\_Country and matchyear>=1991 and matchyear<=2000 group by winner order by matches\_won desc limit 1

union all SELECT winner, count(winner) as matches\_won, '2001-2010' as decade FROM cricket where winner!=Host\_Country and matchyear>=2001 and matchyear<=2010 group by winner order by matches\_won desc limit 1

union all SELECT winner, count(winner) as matches\_won, '2011-2018' as decade FROM cricket where winner!=Host\_Country and matchyear>=2011 and matchyear<=2018 group by winner order by matches\_won desc limit 1)

AS cricket ORDER BY cricket.decade ASC

SELECT winner, count(\*) AS first\_innings\_wins FROM cricket

WHERE (winner == team1 AND innings\_t1 == "First") OR (winner == team2 AND innings\_t2 == "First")

GROUP BY winner

ORDER BY first\_innings\_wins DESC

LIMIT 1

SELECT \* from (select \* from (select margin,winner,count(winner) wins from cricket group by margin,winner order by wins desc)q1 where margin='Winner1stInning' limit 1

union all select \* from (select margin,winner,count(winner) wins from cricket group by margin,winner order by wins desc)q1 where margin='Winner2ndInning' limit 1 )

AS odi2 ORDER by odi2.wins desc limit 1;

SELECT winner , count(winner) as wincount , margin FROM cricket where margin like '%1st%' group by winner,margin ORDER by wincount desc limit 1

union all SELECT winner , count(winner) as wincount , margin FROM cricket where margin like '%2nd%' group by winner,margin ORDER by wincount desc limit 1

create table temp\_drivers (col\_value string);

LOAD DATA INPATH '/user/cloudera/drivers.csv' OVERWRITE INTO TABLE temp\_drivers;

CREATE TABLE drivers (driverID INT,name STRING,ssn BIGINT,location STRING,certified STRING,wageplan STRING)

insert overwrite table drivers

SELECT

regexp\_extract(col\_value, '^(?:([^,]\*),?){1}',1) driverID,

regexp\_extract(col\_value, '^(?:([^,]\*),?){2}',1) name,

regexp\_extract(col\_value, '^(?:([^,]\*),?){3}',1) ssn,

regexp\_extract(col\_value, '^(?:([^,]\*),?){4}',1) location,

regexp\_extract(col\_value, '^(?:([^,]\*),?){5}',1) certified,

regexp\_extract(col\_value, '^(?:([^,]\*),?){6}',1) wageplan

FROM temp\_drivers

create table temp\_timesheet (col\_value string);

LOAD DATA INPATH '/user/cloudera/timesheet.csv' OVERWRITE INTO TABLE temp\_timesheet;

CREATE TABLE timesheet (driverID INT,week INT,hours\_logged INT,miles\_logged INT);

insert overwrite table timesheet

SELECT

regexp\_extract(col\_value, '^(?:([^,]\*),?){1}',1) driverID,

regexp\_extract(col\_value, '^(?:([^,]\*),?){2}',1) week,

regexp\_extract(col\_value, '^(?:([^,]\*),?){3}',1) hours\_logged,

regexp\_extract(col\_value, '^(?:([^,]\*),?){4}',1) miles\_logged

FROM temp\_timesheet

SELECT driverId, sum(hours\_logged), sum(miles\_logged) FROM timesheet GROUP BY driverId;

SELECT d.driverId, d.name, t.total\_hours, t.total\_miles from drivers d

JOIN (SELECT driverId, sum(hours\_logged)total\_hours, sum(miles\_logged)total\_miles FROM timesheet GROUP BY driverId ) t

ON (d.driverId = t.driverId)

create table temp\_stud (col\_value string)

LOAD DATA INPATH '/user/cloudera/studdata.csv' OVERWRITE INTO TABLE temp\_stud;

--drop TABLE stud;

--drop TABLE temp\_stud;

CREATE TABLE stud (gender STRING,NationalITy STRING,POB STRING,stageid STRING,gradeid STRING,sectionid STRING,topic STRING,semester STRING,relation STRING,raisedhands INT,visitedresources INT,announcements INT,discussion INT,ParentAnsweringSurvey STRING,ParentschoolSatisfaction STRING,StudentAbsenceDays STRING,Class STRING)

select \* from temp\_stud limit 100

select \* from stud

insert overwrite table stud

SELECT

regexp\_extract(col\_value, '^(?:([^,]\*),?){1}',1) gender,

regexp\_extract(col\_value, '^(?:([^,]\*),?){2}',1) NationalITy,

regexp\_extract(col\_value, '^(?:([^,]\*),?){3}',1) POB,

regexp\_extract(col\_value, '^(?:([^,]\*),?){4}',1) stageid,

regexp\_extract(col\_value, '^(?:([^,]\*),?){5}',1) gradeid,

regexp\_extract(col\_value, '^(?:([^,]\*),?){6}',1) sectionid,

regexp\_extract(col\_value, '^(?:([^,]\*),?){7}',1) topic,

regexp\_extract(col\_value, '^(?:([^,]\*),?){8}',1) semester,

regexp\_extract(col\_value, '^(?:([^,]\*),?){9}',1) relation,

regexp\_extract(col\_value, '^(?:([^,]\*),?){10}',1) raisedhands,

regexp\_extract(col\_value, '^(?:([^,]\*),?){11}',1) visitedresources,

regexp\_extract(col\_value, '^(?:([^,]\*),?){12}',1) announcements,

regexp\_extract(col\_value, '^(?:([^,]\*),?){13}',1) discussion,

regexp\_extract(col\_value, '^(?:([^,]\*),?){14}',1) ParentAnsweringSurvey,

regexp\_extract(col\_value, '^(?:([^,]\*),?){15}',1) ParentschoolSatisfaction,

regexp\_extract(col\_value, '^(?:([^,]\*),?){16}',1) StudentAbsenceDays,

regexp\_extract(col\_value, '^(?:([^,]\*),?){17}',1) Class

FROM temp\_stud

ALTER TABLE stud SET TBLPROPERTIES ("skip.header.line.count"="1");

select \* from stud;

select topic,stageid,MAX(raisedhands) as x from stud group by topic,stageid order by x desc limit 3;

select gender, NationalITy,stageid,avg(INT(substring(gradeid,4))) as y FROM stud group by gender,NationalITy,stageid order by y desc limit 1;

create table temp\_advertise (col\_value string);

LOAD DATA INPATH '/user/cloudera/advertising.csv' OVERWRITE INTO TABLE temp\_advertise;

--drop table temp\_advertise

--drop table advertise

CREATE TABLE advertise (Timespent float,age INT,areaincome FLOAT,dailyinternetusage FLOAT, adtopiclane STRING, city STRING,male INT,country STRING,timestamp1 timestamp,clickedonad INT)

select \* from advertise limit 100

select \* from advertise

insert overwrite table advertise

SELECT

regexp\_extract(col\_value, '^(?:([^,]\*),?){1}',1) Timespent,

regexp\_extract(col\_value, '^(?:([^,]\*),?){2}',1) age,

regexp\_extract(col\_value, '^(?:([^,]\*),?){3}',1) areaincome,

regexp\_extract(col\_value, '^(?:([^,]\*),?){4}',1) dailyinternetusage,

regexp\_extract(col\_value, '^(?:([^,]\*),?){5}',1) adtopiclane,

regexp\_extract(col\_value, '^(?:([^,]\*),?){6}',1) city,

regexp\_extract(col\_value, '^(?:([^,]\*),?){7}',1) male,

regexp\_extract(col\_value, '^(?:([^,]\*),?){8}',1) country,

regexp\_extract(col\_value, '^(?:([^,]\*),?){9}',1) timestamp1,

regexp\_extract(col\_value, '^(?:([^,]\*),?){10}',1) clickedonad

FROM temp\_advertise

ALTER TABLE advertise SET TBLPROPERTIES ("skip.header.line.count"="1");

select count(clickedonad) as clicks from advertise where clickedonad=1;

select count(clickedonad) as clicks from advertise where male=0 and clickedonad=1;

select count(clickedonad) as clicks from advertise where male=1 and clickedonad=1;

SELECT title FROM movies ORDER BY title ASC LIMIT 5 OFFSET 5;

SELECT title, domestic\_sales, international\_sales FROM movies JOIN boxofficemON movies.id = boxoffice.movie\_id;