1. Upload hr\_db dataset into HDFS

$ hadoop fs -put /home/hduser/Downloads/sharedfolder/hr\_db /user/hduser/

2. Solve the following use cases using Hive SQL or Spark RDD or Spark DF (Pick any one of these)

a. Retrieve last\_name, first\_name, salary, salary+300 as increment salary from employees table

b. Find out the ID's of departments in which employees are working

c. Retrieve last\_name, job\_id, department\_id of employee having last name as Whalen

d. Retrieve last name and salary of all employees who have salary greater than 5000 working in department 90

e. Retrieve last\_name, salary of all employees working as 'SA\_REP','AD\_PRES' earning above 5000, sort the data in ascending order of last name

f. Retrieve the last names of employees whose last but one character of the last name is e

g. Retrieve the names of employees not having managers

h. Retrieve the employee name and the name of the department in which the employee is working

I. List all the department ids having SIX employees

**SOLUTIONS**:

1. Upload hr\_db dataset into HDFS

$ hadoop fs -put /home/hduser/Downloads/sharedfolder/hr\_db /user/hduser/

(hadoop fs -rmr /user/hduser/hr\_db; - To remove the files from hdfs)

hive > CREATE DATABASE employees\_demo\_db;

hive > CREATE EXTERNAL TABLE employees(

emp\_id INT,

first\_name STRING,

last\_name STRING,

short\_name STRING,

location STRING,

dob STRING,

job\_code STRING,

salary DOUBLE,

manager\_id DOUBLE,

job\_id INT,

department\_id INT,

jd\_id STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

LOCATION '/user/hduser/hr\_db/employees' ;

1. Solve the following use cases using Hive SQL or Spark RDD or Spark DF (Pick any one of these)
2. Retrieve last\_name, first\_name, salary, salary+300 as increment salary from employees table

Solution :

SET hive.cli.print.header=true;

hive > SELECT last\_name, first\_name, salary, salary + 300 AS increment\_salary FROM employees;

last\_name first\_name salary increment\_salary

King Steven 24000.0 24300.0

Kochhar Neena 17000.0 17300.0

De Haan Lex 17000.0 17300.0

Hunold Alexander 9000.0 9300.0

Ernst Bruce 6000.0 6300.0

Austin David 4800.0 5100.0

Pataballa Valli 4800.0 5100.0

Lorentz Diana 4200.0 4500.0

Greenberg Nancy 12000.0 12300.0

Faviet Daniel 9000.0 9300.0

Chen John 8200.0 8500.0

Sciarra Ismael 7700.0 8000.0

Urman Jose Manuel 7800.0 8100.0

Popp Luis 6900.0 7200.0

Raphaely Den 11000.0 11300.0

Khoo Alexander 3100.0 3400.0

Baida Shelli 2900.0 3200.0

Tobias Sigal 2800.0 3100.0

Himuro Guy 2600.0 2900.0

Colmenares Karen 2500.0 2800.0

Weiss Matthew 8000.0 8300.0

Fripp Adam 8200.0 8500.0

Kaufling Payam 7900.0 8200.0

Vollman Shanta 6500.0 6800.0

Mourgos Kevin 5800.0 6100.0

Nayer Julia 3200.0 3500.0

Mikkilineni Irene 2700.0 3000.0

Landry James 2400.0 2700.0

Markle Steven 2200.0 2500.0

Bissot Laura 3300.0 3600.0

Atkinson Mozhe 2800.0 3100.0

Marlow James 2500.0 2800.0

Olson TJ 2100.0 2400.0

Mallin Jason 3300.0 3600.0

Rogers Michael 2900.0 3200.0

Gee Ki 2400.0 2700.0

Philtanker Hazel 2200.0 2500.0

Ladwig Renske 3600.0 3900.0

Stiles Stephen 3200.0 3500.0

Seo John 2700.0 3000.0

Patel Joshua 2500.0 2800.0

Rajs Trenna 3500.0 3800.0

Davies Curtis 3100.0 3400.0

Matos Randall 2600.0 2900.0

Vargas Peter 2500.0 2800.0

Russell John 14000.0 14300.0

Partners Karen 13500.0 13800.0

Errazuriz Alberto 12000.0 12300.0

Cambrault Gerald 11000.0 11300.0

Zlotkey Eleni 10500.0 10800.0

Tucker Peter 10000.0 10300.0

Bernstein David 9500.0 9800.0

Hall Peter 9000.0 9300.0

Olsen Christopher 8000.0 8300.0

Cambrault Nanette 7500.0 7800.0

Tuvault Oliver 7000.0 7300.0

King Janette 10000.0 10300.0

Sully Patrick 9500.0 9800.0

McEwen Allan 9000.0 9300.0

Smith Lindsey 8000.0 8300.0

Doran Louise 7500.0 7800.0

Sewall Sarath 7000.0 7300.0

Vishney Clara 10500.0 10800.0

Greene Danielle 9500.0 9800.0

Marvins Mattea 7200.0 7500.0

Lee David 6800.0 7100.0

Ande Sundar 6400.0 6700.0

Banda Amit 6200.0 6500.0

Ozer Lisa 11500.0 11800.0

Bloom Harrison 10000.0 10300.0

Fox Tayler 9600.0 9900.0

Smith William 7400.0 7700.0

Bates Elizabeth 7300.0 7600.0

Kumar Sundita 6100.0 6400.0

Abel Ellen 11000.0 11300.0

Hutton Alyssa 8800.0 9100.0

Taylor Jonathon 8600.0 8900.0

Livingston Jack 8400.0 8700.0

Grant Kimberely 7000.0 7300.0

Johnson Charles 6200.0 6500.0

Taylor Winston 3200.0 3500.0

Fleaur Jean 3100.0 3400.0

Sullivan Martha 2500.0 2800.0

Geoni Girard 2800.0 3100.0

Sarchand Nandita 4200.0 4500.0

Bull Alexis 4100.0 4400.0

Dellinger Julia 3400.0 3700.0

Cabrio Anthony 3000.0 3300.0

Chung Kelly 3800.0 4100.0

Dilly Jennifer 3600.0 3900.0

Gates Timothy 2900.0 3200.0

Perkins Randall 2500.0 2800.0

Bell Sarah 4000.0 4300.0

Everett Britney 3900.0 4200.0

McCain Samuel 3200.0 3500.0

Jones Vance 2800.0 3100.0

Walsh Alana 3100.0 3400.0

Feeney Kevin 3000.0 3300.0

OConnell Donald 2600.0 2900.0

Grant Douglas 2600.0 2900.0

Whalen Jennifer 4400.0 4700.0

Hartstein Michael 13000.0 13300.0

Fay Pat 6000.0 6300.0

Mavris Susan 6500.0 6800.0

Baer Hermann 10000.0 10300.0

Higgins Shelley 12000.0 12300.0

Gietz William 8300.0 8600.0

Time taken: 0.149 seconds, Fetched: 107 row(s)

1. Find out the ID's of departments in which employees are working

Solution :

SELECT DISTINCT department\_id FROM employees ;

NULL

10

20

30

40

50

60

70

80

90

100

110

Time taken: 18.745 seconds, Fetched: 12 row(s)

1. Retrieve last\_name, job\_id, department\_id of employee having last name as Whalen

Solution :

SELECT last\_name, job\_id, department\_id FROM employees WHERE last\_name = 'Whalen';

Whalen 101 10

Time taken: 0.136 seconds, Fetched: 1 row(s)

1. Retrieve last name and salary of all employees who have salary greater than 5000 working in department 90

Solution:

SELECT last\_name, salary FROM employees WHERE salary > 5000 AND department\_id=90;

King 24000.0

Kochhar 17000.0

De Haan 17000.0

Time taken: 1.584 seconds, Fetched: 3 row(s)

1. Retrieve last\_name, salary of all employees working as 'SA\_REP','AD\_PRES' earning above 5000, sort the data in ascending order of last name

Solution:

SELECT last\_name, salary FROM employees WHERE job\_code IN ('SA\_REP','AD\_PRES') AND salary > 5000 ORDER BY last\_name;

Abel 11000.0

Ande 6400.0

Banda 6200.0

Bates 7300.0

Bernstein 9500.0

Bloom 10000.0

Cambrault 7500.0

Doran 7500.0

Fox 9600.0

Grant 7000.0

Greene 9500.0

Hall 9000.0

Hutton 8800.0

Johnson 6200.0

King 24000.0

King 10000.0

Kumar 6100.0

Lee 6800.0

Livingston 8400.0

Marvins 7200.0

McEwen 9000.0

Olsen 8000.0

Ozer 11500.0

Sewall 7000.0

Smith 8000.0

Smith 7400.0

Sully 9500.0

Taylor 8600.0

Tucker 10000.0

Tuvault 7000.0

Vishney 10500.0

Time taken: 18.23 seconds, Fetched: 31 row(s)

1. Retrieve the last names of employees whose last but one character of the last name is e

Solution:

SELECT last\_name FROM employees WHERE last\_name LIKE '%\_e\_';

Faviet

Chen

Colmenares

Nayer

Gee

Philtanker

Stiles

Seo

Patel

Davies

Zlotkey

Tucker

Olsen

McEwen

Vishney

Lee

Ozer

Bates

Abel

Dellinger

Gates

Jones

Feeney

Whalen

Baer

Time taken: 0.069 seconds, Fetched: 25 row(s)

1. Retrieve the names of employees not having managers

Solution:

SELECT \* FROM employees WHERE manager\_id = NULL;

1. Retrieve the employee name and the name of the department in which the employee is working

Solution :

SELECT first\_name,last\_name,job\_code FROM employees;

Steven King AD\_PRES

Neena Kochhar AD\_VP

Lex De Haan AD\_VP

Alexander Hunold IT\_PROG

Bruce Ernst IT\_PROG

David Austin IT\_PROG

Valli Pataballa IT\_PROG

Diana Lorentz IT\_PROG

Nancy Greenberg FI\_MGR

Daniel Faviet FI\_ACCOUNT

John Chen FI\_ACCOUNT

Ismael Sciarra FI\_ACCOUNT

Jose Manuel Urman FI\_ACCOUNT

Luis Popp FI\_ACCOUNT

Den Raphaely PU\_MAN

Alexander Khoo PU\_CLERK

Shelli Baida PU\_CLERK

Sigal Tobias PU\_CLERK

Guy Himuro PU\_CLERK

Karen Colmenares PU\_CLERK

Matthew Weiss ST\_MAN

Adam Fripp ST\_MAN

Payam Kaufling ST\_MAN

Shanta Vollman ST\_MAN

Kevin Mourgos ST\_MAN

Julia Nayer ST\_CLERK

Irene Mikkilineni ST\_CLERK

James Landry ST\_CLERK

Steven Markle ST\_CLERK

Laura Bissot ST\_CLERK

Mozhe Atkinson ST\_CLERK

James Marlow ST\_CLERK

TJ Olson ST\_CLERK

Jason Mallin ST\_CLERK

Michael Rogers ST\_CLERK

Ki Gee ST\_CLERK

Hazel Philtanker ST\_CLERK

Renske Ladwig ST\_CLERK

Stephen Stiles ST\_CLERK

John Seo ST\_CLERK

Joshua Patel ST\_CLERK

Trenna Rajs ST\_CLERK

Curtis Davies ST\_CLERK

Randall Matos ST\_CLERK

Peter Vargas ST\_CLERK

John Russell SA\_MAN

Karen Partners SA\_MAN

Alberto Errazuriz SA\_MAN

Gerald Cambrault SA\_MAN

Eleni Zlotkey SA\_MAN

Peter Tucker SA\_REP

David Bernstein SA\_REP

Peter Hall SA\_REP

Christopher Olsen SA\_REP

Nanette Cambrault SA\_REP

Oliver Tuvault SA\_REP

Janette King SA\_REP

Patrick Sully SA\_REP

Allan McEwen SA\_REP

Lindsey Smith SA\_REP

Louise Doran SA\_REP

Sarath Sewall SA\_REP

Clara Vishney SA\_REP

Danielle Greene SA\_REP

Mattea Marvins SA\_REP

David Lee SA\_REP

Sundar Ande SA\_REP

Amit Banda SA\_REP

Lisa Ozer SA\_REP

Harrison Bloom SA\_REP

Tayler Fox SA\_REP

William Smith SA\_REP

Elizabeth Bates SA\_REP

Sundita Kumar SA\_REP

Ellen Abel SA\_REP

Alyssa Hutton SA\_REP

Jonathon Taylor SA\_REP

Jack Livingston SA\_REP

Kimberely Grant SA\_REP

Charles Johnson SA\_REP

Winston Taylor SH\_CLERK

Jean Fleaur SH\_CLERK

Martha Sullivan SH\_CLERK

Girard Geoni SH\_CLERK

Nandita Sarchand SH\_CLERK

Alexis Bull SH\_CLERK

Julia Dellinger SH\_CLERK

Anthony Cabrio SH\_CLERK

Kelly Chung SH\_CLERK

Jennifer Dilly SH\_CLERK

Timothy Gates SH\_CLERK

Randall Perkins SH\_CLERK

Sarah Bell SH\_CLERK

Britney Everett SH\_CLERK

Samuel McCain SH\_CLERK

Vance Jones SH\_CLERK

Alana Walsh SH\_CLERK

Kevin Feeney SH\_CLERK

Donald OConnell SH\_CLERK

Douglas Grant SH\_CLERK

Jennifer Whalen AD\_ASST

Michael Hartstein MK\_MAN

Pat Fay MK\_REP

Susan Mavris HR\_REP

Hermann Baer PR\_REP

Shelley Higgins AC\_MGR

William Gietz AC\_ACCOUNT

Time taken: 0.104 seconds, Fetched: 107 row(s)

1. List all the department ids having SIX employees

Solution:

SELECT department\_id,COUNT(\*) FROM employees GROUP BY department\_id HAVING COUNT(\*) >6;

50 45

80 34

Time taken: 18.044 seconds, Fetched: 2 row(s)