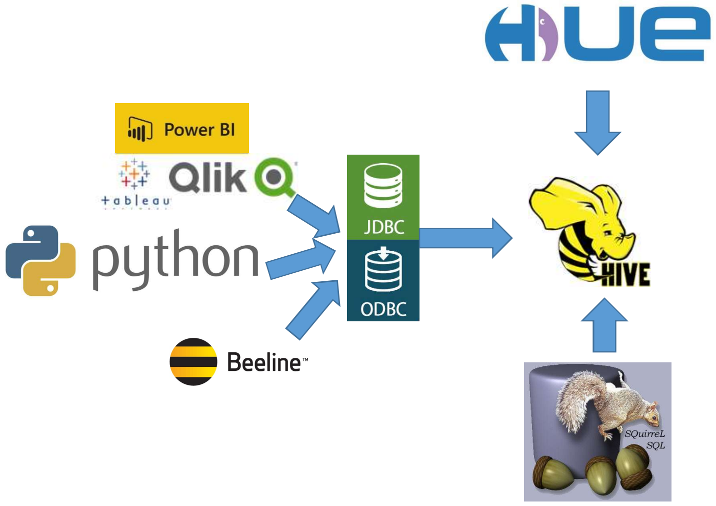
Apache Hive

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

© Dr Heni Bouhamed
Big Data Trainer
Senior Lecturer at Sfax University
Senior Lecturer at ESTYA University (France)
Cloudera Instructor at Elitech Paris
Certified CDOSS Big Data Instructor
Heni.bouhamed@fsegs.usf.tn

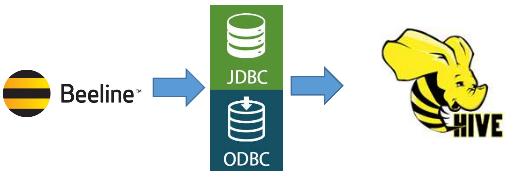
CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive



Interactive execution :
$ beeline –u jdbc:hive2://
-n (user name) u1
-p (password) hadoop
!quit to exit (!q also)
/databasesname

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q1: Once you're in the query editor in Hive or Beeline, Which of the following is a way to get the schema of the makers table?

- Execute the command use toy; to make toy database as the active database and then run the command DESCRIBE makers;
- Select the toy database as the active database and then run the command SHOW TABLE makers;
- Select the toy database as the active database and then run the command SHOW TABLES;
- Select the toy database as the active database and then run the command SHOW DATABASE;
- Select the toy database as the active database and then run the command DESCRIBE TABLES;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q2: Which of the following provide a user interface where you can enter and run SQL queries? Check all that apply.

- Tableau
- Hue
- Linux
- Beeline
- ODBC

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q3: Which shell uses JDBC to connect to its query engine?

- Beeline
- Hive

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q4: Which is a required argument for the beeline command on the VM?

- -u for user
- -u for URL
- -c for connect
- -n for name
- -p for password
- -c for command

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

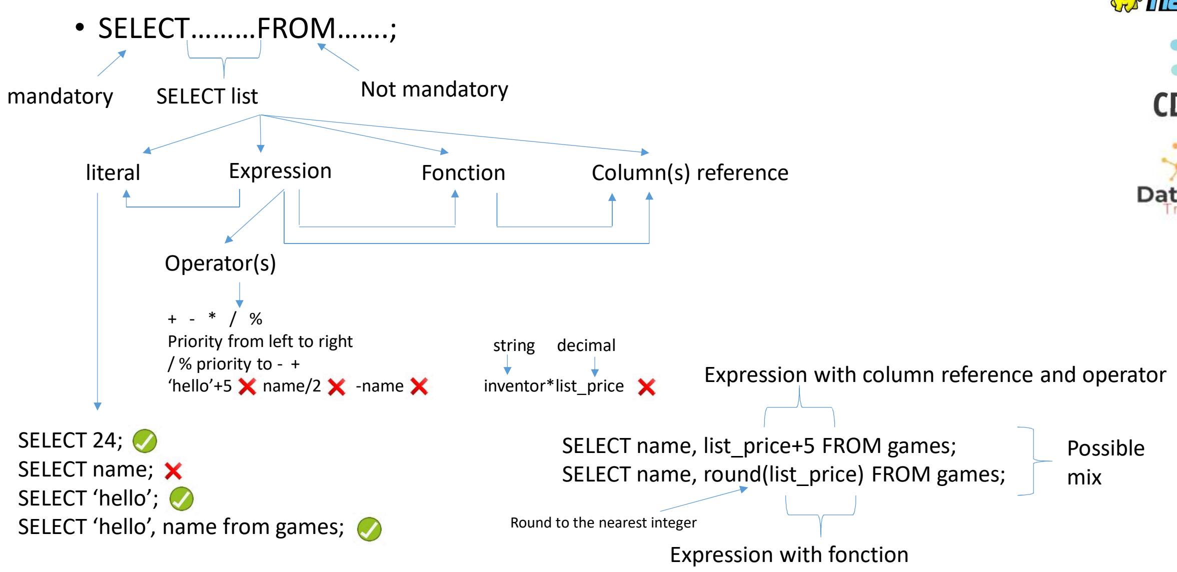
Apache Hive

Q5: Which are valid statements or commands once you are in Beeline? Check all that apply.

- SELECT * from tablename;
- SELECT * from tablename
- !quit;
- !quit

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive



- SELECT………FROM…….;

mandatory

SELECT list

Not mandatory

literal

Expression

Fonction

Column(s) reference

Operator(s)

+ - * / %

Priority from left to right

/ % priority to - +

‘hello’+5 ✗ name/2 ✗ -name ✗

string decimal

inventor*list_price ✗

Expression with column reference and operator

SELECT 24; ✓

SELECT name; ✗

SELECT ‘hello’; ✓

SELECT ‘hello’, name from games; ✓

SELECT name, list_price+5 FROM games;

SELECT name, round(list_price) FROM games;

Possible mix

Round to the nearest integer

Expression with fonction

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

- **Data types**

- Numeric

Integer Data Types
- tinyint: -128 to 127
- smallint: -32768 to 32767
- Int: -2147483648 to 2147483647
- Bigint: -9.2 quintillion to 9.2 quintillion

- Decimal data types
- Float
- Double
- Decimal

- Character
- String
- Char
- Varchar

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

- **Alias**

Execute the following query with Hive :
Select name, 5, list_price+5 from games;

• You can force the name of a column with AS (optional) :
Select name, 5 as shipping_fee, list_price+5 as price_with_shipping from games;

• Alias can be a mix of lettre, digits, underscores

• Alias can’t be only digits or a specific word

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

- **Built in functions (Hive)**

- Lowercase by convention
- Some mathematical functions:

round(16,39)=16
round(16.39,1)=16.4
round(4.5)=5
round(-4.5)=-5
floor(19.37)=19
ceil(19.37)=20
pow(2,3)=8
abs(-9)=9
sqrt(4)=2

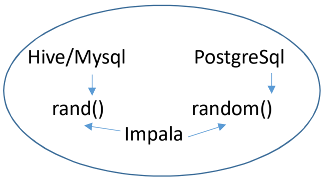
- Some String manipulation functions :

length(str)
reverse(str)
upper(str)
concat(str1,str2)
concat_ws(sep,str1,str2,…) etc…

- Some aggregation functions :

max(col reference)
min(col reference)
count(*)

Attention!



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

▪ Casting

Select concat(name, ‘is for’, min_age, ‘or older’ from games;

Hive does the casting implicitly

But it is better to caste min_age with cast(min_age as string)

▪ Distinct

Select distinct min_age from games; only once at the beginning, ALL is its opposite .
Possible with several columns, possible with * and possible with functions

Select distinct min_age, max_players
FROM games;

min_age	Max_players
8	6
8	4
8	6
3	4
10	5

min_age	Max_players
8	6
8	4
3	4
10	5

Select distinct
concat(substring(year,1,3), «0s»)
FROM games;

Concat(substring(year,1,3), « 0s »)
1900s
1950s
1930s
1940s

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

▪ From

From database.table → replace use databases;

- Data bases identifiers

From database.table → replace use databases;

▪ Identifiers possibilities

- Letters, digits, underscores
- Letter for first character
- Lowercase letters
- Max length varies (recommend fewer than 30)

▪ Some examples of reserved words

- FROM
- AS
- DISTINCT
- SHOW
- USE

www.tiny.cloudera.com/hive-reserved-words
You can use them with back quote:

```
USE `use`;
SELECT `select` FROM `from`;
```

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

- **Beeline in non-interactive mode**

Beeline -u … -e ‘select…; select…; !quit;’
Beeline -u … -f myquery.sql
--silent=true (before others)
To choose a database :

- Beeline –u …/db
- From from db.table
- Use db; in query

- Change output format using - -outputformat=:
  - csv2 for comma delimited
  - Tsv2 for tab delimited
- Exclude header using - -showHeader=false

```
Beeline --showHeader=false --outputformat=csv2            \
-u jdbc:hive2:// -e ‘select id, name from fun.games; !quit;’ > games.csv
```

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q1: Consider this statement:

- SELECT name, min_players, max_players FROM games;

Which part of this statement is the SELECT list?

- SELECT
- FROM
- games
- FROM games
- SELECT name, min_players, max_players FROM games; (that is, the whole statement)
- name, min_players, max_players
- SELECT name, min_players, max_players

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q2: The table toys has columns id (integer), name (string), price (decimal), and maker_id (integer) which of the following statements is the SELECT list a literal string? Check all that apply.

- SELECT Lite-Brite FROM toys
- SELECT 'toys'
- SELECT * FROM toys
- SELECT 'name' FROM toys
- SELECT toys
- SELECT name FROM toys

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q3: The table toys has columns id (integer), name (string), price (decimal), and maker_id (integer). Which of the following are valid SELECT statements? Check all that apply. (You might want to use the VM and try them!)

SELECT name, price FROM toys;

SELECT toys;

SELECT 1000;

SELECT FROM toys: name, price;

SELECT FROM toys COLUMNS name, price;

SELECT 'Lite-Brite';

SELECT toys.name, toys.price;

SELECT * FROM toys;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q4: The following expression will cause an error when used in a SELECT statement:

"-7.5" % 3.2

What is the error? (You might want to use the VM and try it!)

- The data types are incompatible.
- The indicated operation only works with integers.
- The indicated operation only works with negative numbers.
- An invalid operator symbol is used.

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q5: The profit for an item sold can be found using the formula profit = price – cost. You want to query an inventory table with columns name, price, and cost, and get the following result:

name	profit
Widget 38	15
Widget 38e	49
Gadget 2000	72

Which SELECT statements will produce that table? Check all that apply.

- SELECT name, price - cost, profit FROM inventory
- SELECT name, price - cost profit FROM inventory
- SELECT name, price - cost AS profit FROM inventory
- SELECT name, price - cost (AS profit) FROM inventory
- SELECT name, price - cost FROM inventory
- SELECT name, profit FROM inventory

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q6: What is the result of the following SELECT statement?

SELECT round(3.47) ;

Q8: What is the result of the following SELECT statement?

SELECT ceil(-2.47);

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q7: Which SELECT statement will give 3 * 3 * 3 * 3 in Hive?

- SELECT 3^4;
- SELECT pow(3,4);
- SELECT abs(3,4);
- SELECT round(3,4);

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q8: First, try writing a SELECT statement to answer the question: When will each of the games in the games table turn one hundred years old? In other words, what year will mark the one hundredth anniversary of the invention of each game? Write a SELECT statement that answers this question, and run it with Hive. You should include the names of the games in the result set. Then answer the following question:

Which game has already had its 100th anniversary?

- Candy Land
- Scrabble
- Monopoly
- Risk
- Clue

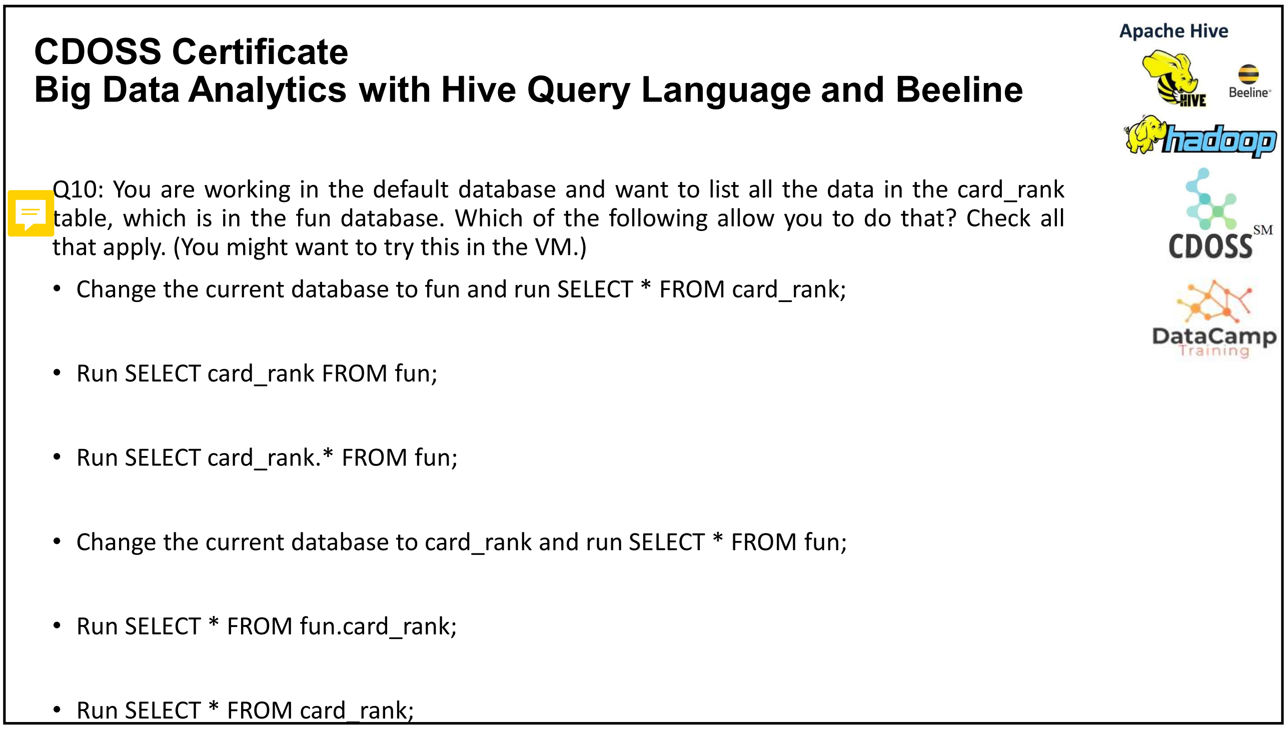
Using SELECT name, cast(year AS INT) + 100 as century_year FROM games; shows that Monopoly turned 100 in 2003.

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q9: How many unique values are there in the min_players column of the fun.games table? Write and run a SQL query to check.

You should get only one row if you run SELECT DISTINCT min_players FROM games;

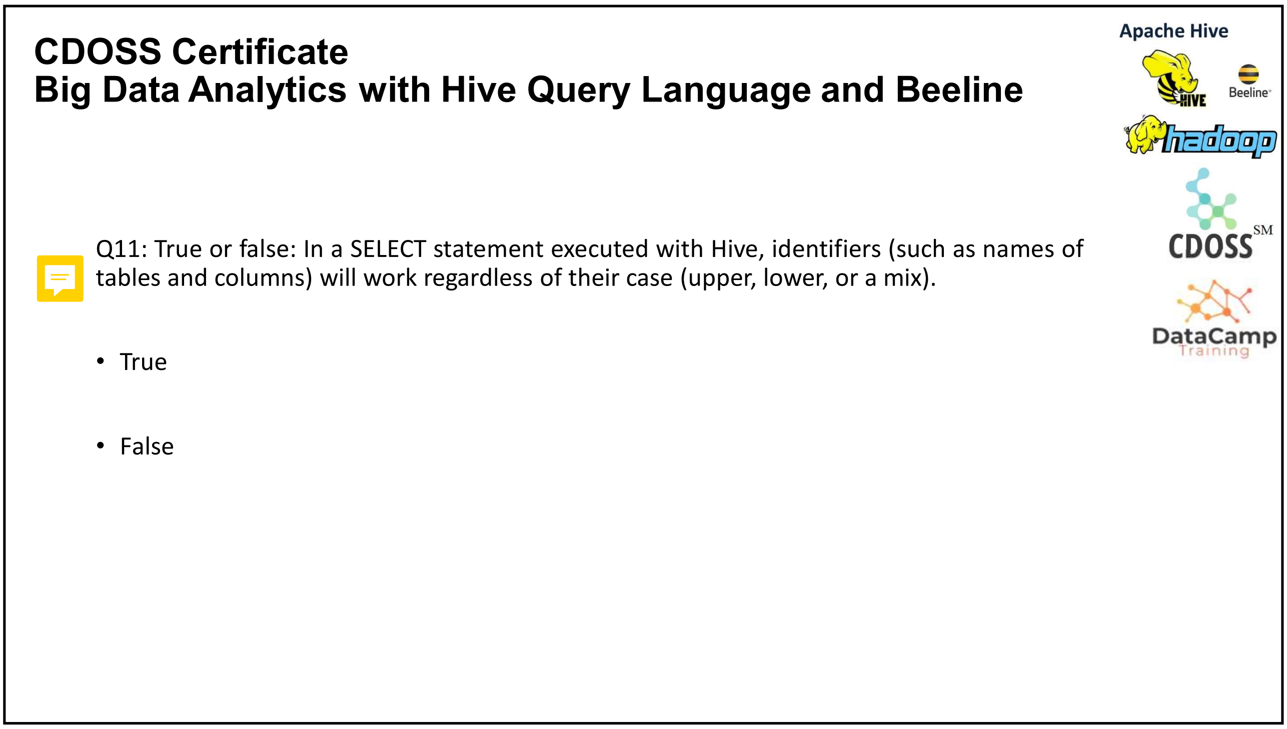


CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q10: You are working in the default database and want to list all the data in the card_rank table, which is in the fun database. Which of the following allow you to do that? Check all that apply. (You might want to try this in the VM.)

- Change the current database to fun and run SELECT * FROM card_rank;
- Run SELECT card_rank FROM fun;
- Run SELECT card_rank.* FROM fun;
- Change the current database to card_rank and run SELECT * FROM fun;
- Run SELECT * FROM fun.card_rank;
- Run SELECT * FROM card_rank;

Apache Hive
HIVE
Beeline
hadoop
CDOSS SM
DataCamp
Training



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q11: True or false: In a SELECT statement executed with Hive, identifiers (such as names of tables and columns) will work regardless of their case (upper, lower, or a mix).

- True
- False

Apache Hive
HIVE
Beeline
hadoop
CDOSS SM
DataCamp
Training

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q12: Which are true of the formatting of SELECT statements (for example, how they would look in the Hive query editor)? Check all that apply.

- Newlines (line or paragraph breaks) can only be added just before a new keyword
- Indenting with a tab character is necessary when a clause is too long for a single line
- Extra spaces are ignored if not in a keyword, identifier, or quoted string
- By convention, indent clauses every time you start them on a new line

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q13: Which is a correct command for running a SQL query through Beeline in a command line argument?

- beeline -u jdbc:hive2:// -e 'SELECT * FROM fun.games; !quit;'
- beeline -u jdbc:hive2:// -r SELECT * FROM fun.games; !quit;
- beeline -u jdbc:hive2:// -r 'SELECT * FROM fun.games; !quit;'
- beeline -u jdbc:hive2:// -f 'SELECT * FROM fun.games; !quit;'
- beeline -u jdbc:hive2:// -e SELECT * FROM fun.games; !quit;
- beeline -u jdbc:hive2:// -f SELECT * FROM fun.games; !quit;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q14: Here are some commands for you to try in the VM. The $ means you should type this at your command line prompt, not within Beeline.

- $ beeline -h
- $ beeline -u jdbc:hive2:// -e 'SELECT * FROM fun.games; !quit;'
- $ beeline -u jdbc:hive2:// -e 'USE fun; SELECT * FROM games; !quit;'
- $ beeline -u jdbc:hive2:// --silent=true -e 'SELECT * FROM fun.games; !quit;'

Before doing more commands, first create a file and put the following queries in it and save it in your current directory (unless you specify otherwise).

- USE fun;
- SELECT * FROM games;
- SELECT name, list_price, 0.8*list_price AS discounted_price FROM games; !quit

Then try these commands. Compare what happens.

- $ beeline -u jdbc:hive2:// -f commands.sql
- $ beeline -u jdbc:hive2:// --silent=true -f commands.sql

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

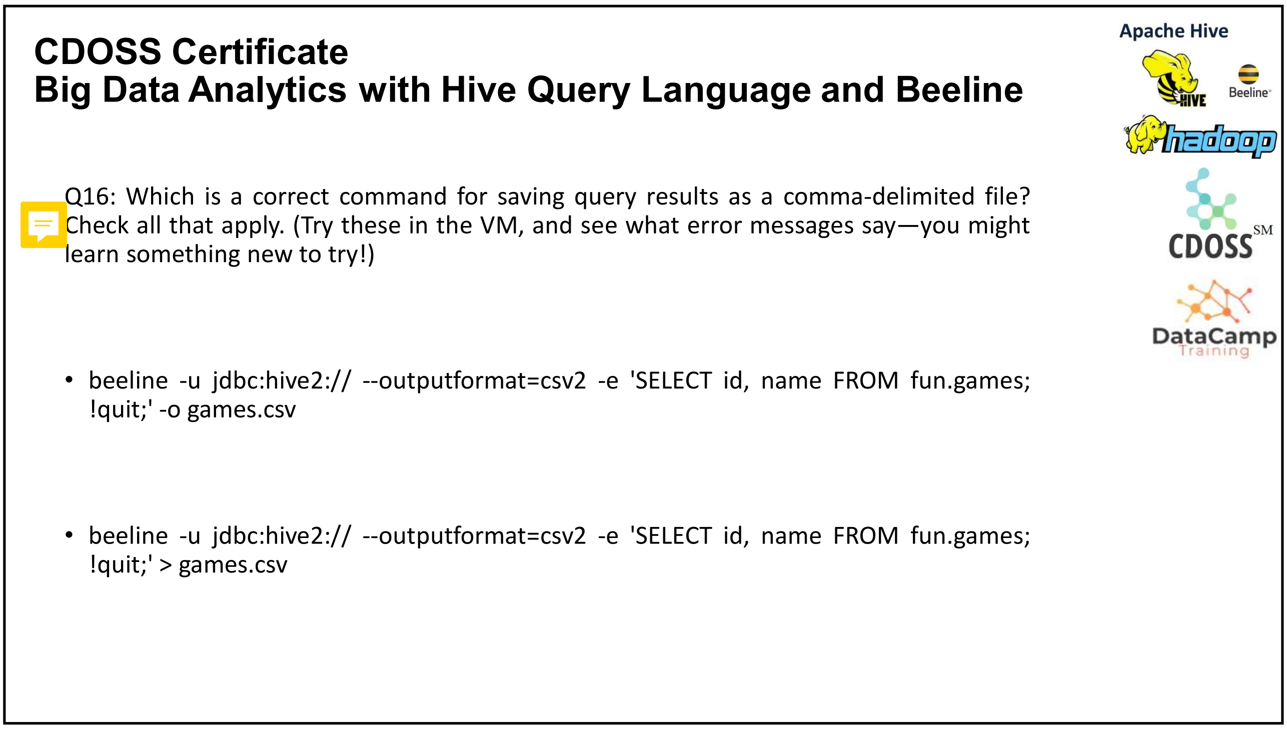
Apache Hive

Q15: Suppose you want your query results to output to the terminal in this format:

```
1,Monopoly,Elizabeth Magie,1903,8,2,6,19.99
2,Scrabble,Alfred Mosher Butts,1938,8,2,4,17.99
3,Clue,Anthony E. Pratt,1944,8,2,6,9.99
4,Candy Land,Eleanor Abbott,1948,3,2,4,7.99
5,Risk,Albert Lamorisse,1957,10,2,5,29.99
```

Which commands will produce this? Check all that apply.

- beeline -u jdbc:hive2:// --outputformat=csv2 --showHeader=false -e 'SELECT * FROM fun.games; !quit;'
- beeline -u jdbc:hive2:// --outputformat=csv2 -e 'SELECT * FROM fun.games; !quit;'
- beeline -u jdbc:hive2:// --outputformat=',' --showHeader=false -e 'SELECT * FROM fun.games; !quit;'
- beeline -u jdbc:hive2:// --outputformat=',' -e 'SELECT * FROM fun.games; !quit;'

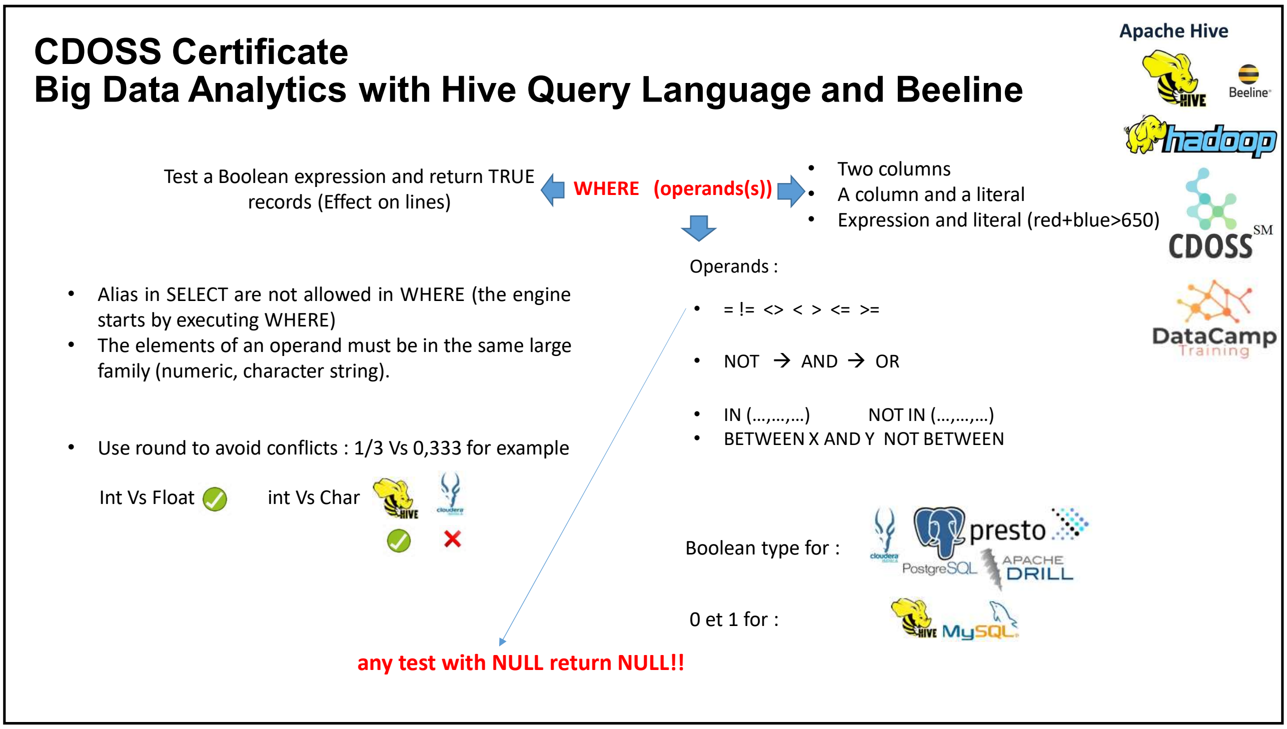


CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive
HIVE
Beeline
hadoop
CDOSS SM
DataCamp Training

Q16: Which is a correct command for saving query results as a comma-delimited file? Check all that apply. (Try these in the VM, and see what error messages say—you might learn something new to try!)

- beeline -u jdbc:hive2:// --outputformat=csv2 -e 'SELECT id, name FROM fun.games; !quit;' -o games.csv
- beeline -u jdbc:hive2:// --outputformat=csv2 -e 'SELECT id, name FROM fun.games; !quit;' > games.csv



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive
HIVE
Beeline
hadoop
CDOSS SM
DataCamp Training

Test a Boolean expression and return TRUE records (Effect on lines) ⇦ **WHERE (operands(s))** ⇨

- Two columns
- A column and a literal
- Expression and literal (red+blue>650)

⇩

Operands :

- = != <> < > <= >=
- NOT → AND → OR
- IN (…,…,…) NOT IN (…,…,…)
- BETWEEN X AND Y NOT BETWEEN

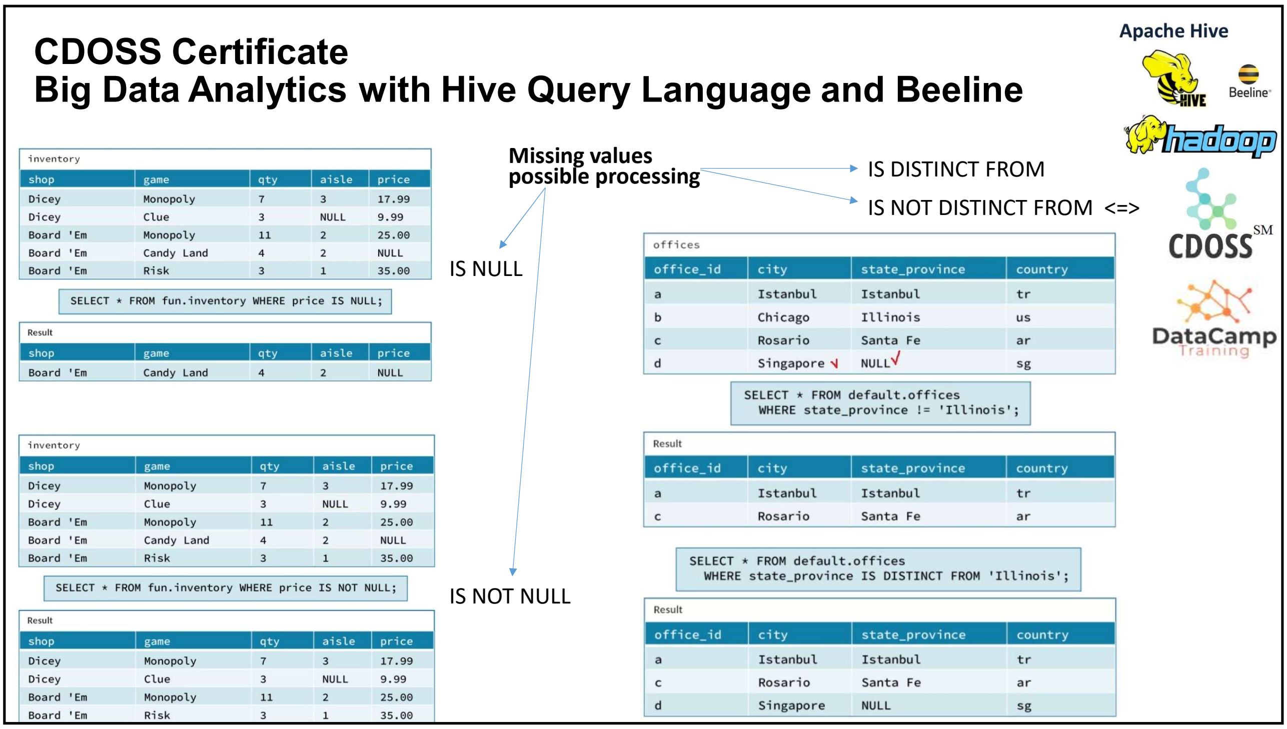
- Alias in SELECT are not allowed in WHERE (the engine starts by executing WHERE)
- The elements of an operand must be in the same large family (numeric, character string).
- Use round to avoid conflicts : 1/3 Vs 0,333 for example

Int Vs Float ✓ int Vs Char HIVE ✓ cloudera ✗

Boolean type for :

0 et 1 for :

any test with NULL return NULL!!



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

inventory

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00

```
SELECT * FROM fun.inventory WHERE price IS NULL;
```

Result

shop	game	qty	aisle	price
Board 'Em	Candy Land	4	2	NULL

inventory

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00

```
SELECT * FROM fun.inventory WHERE price IS NOT NULL;
```

Result

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Risk	3	1	35.00

Missing values possible processing

IS NULL

IS NOT NULL

IS DISTINCT FROM

IS NOT DISTINCT FROM <=>

offices

office_id	city	state_province	country
a	Istanbul	Istanbul	tr
b	Chicago	Illinois	us
c	Rosario	Santa Fe	ar
d	Singapore ✓	NULL ✓	sg

```
SELECT * FROM default.offices
  WHERE state_province != 'Illinois';
```

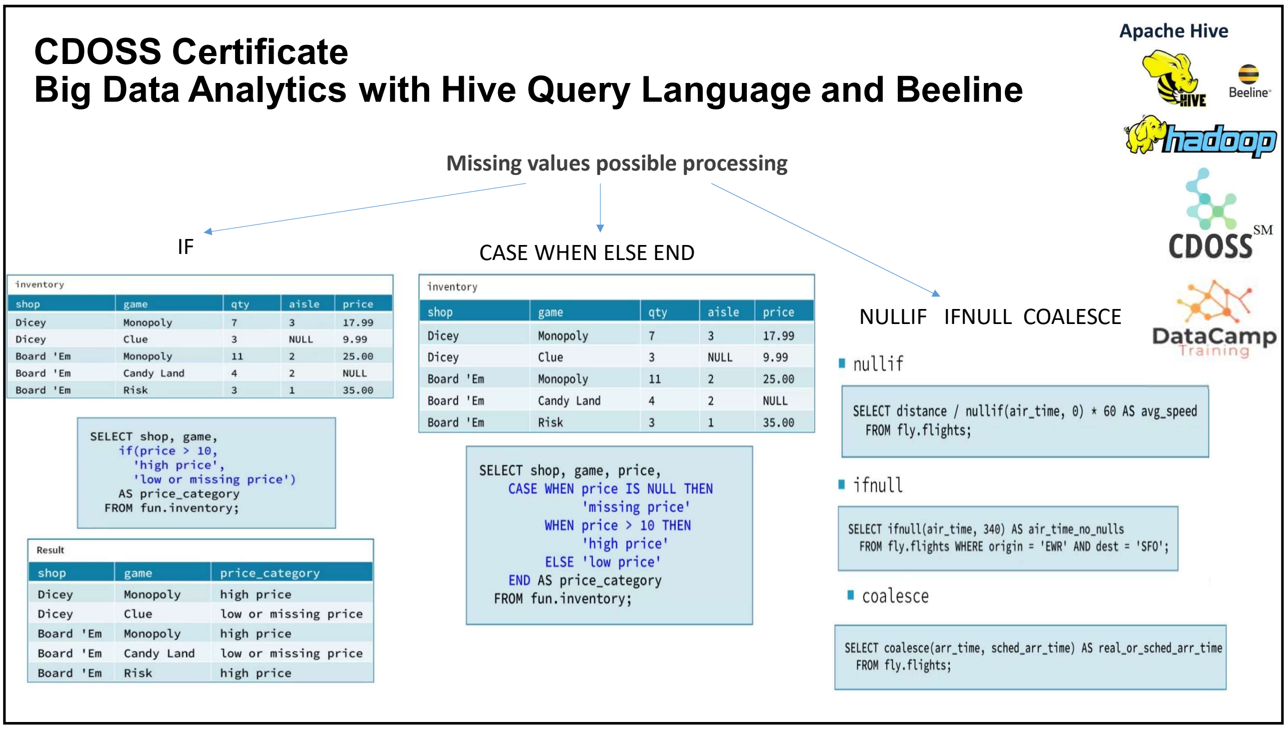
Result

office_id	city	state_province	country
a	Istanbul	Istanbul	tr
c	Rosario	Santa Fe	ar

```
SELECT * FROM default.offices
  WHERE state_province IS DISTINCT FROM 'Illinois';
```

Result

office_id	city	state_province	country
a	Istanbul	Istanbul	tr
c	Rosario	Santa Fe	ar
d	Singapore	NULL	sg



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Missing values possible processing

IF

inventory

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00

```
SELECT shop, game,
    if(price > 10,
      'high price',
      'low or missing price')
    AS price_category
  FROM fun.inventory;
```

Result

shop	game	price_category
Dicey	Monopoly	high price
Dicey	Clue	low or missing price
Board 'Em	Monopoly	high price
Board 'Em	Candy Land	low or missing price
Board 'Em	Risk	high price

CASE WHEN ELSE END

inventory

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00

```
SELECT shop, game, price,
    CASE WHEN price IS NULL THEN
          'missing price'
        WHEN price > 10 THEN
          'high price'
        ELSE 'low price'
    END AS price_category
  FROM fun.inventory;
```

NULLIF IFNULL COALESCE

- nullif

```
SELECT distance / nullif(air_time, 0) * 60 AS avg_speed
  FROM fly.flights;
```

- ifnull

```
SELECT ifnull(air_time, 340) AS air_time_no_nulls
  FROM fly.flights WHERE origin = 'EWR' AND dest = 'SFO';
```

- coalesce

```
SELECT coalesce(arr_time, sched_arr_time) AS real_or_sched_arr_time
  FROM fly.flights;
```

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

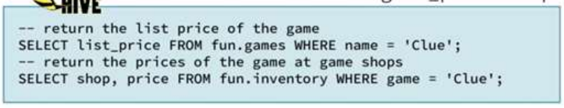
Apache Hive

Beeline Variables → Terminal instanciation

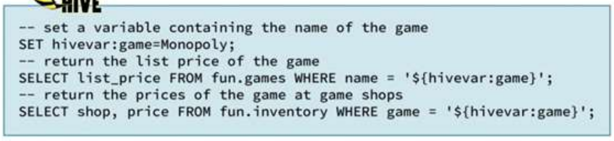
↓

File instanciation

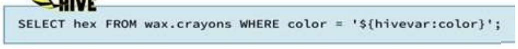
game_prices.sql



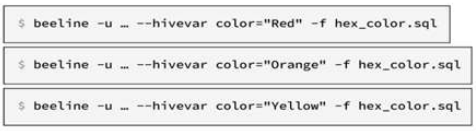
```
-- return the list price of the game
SELECT list_price FROM fun.games WHERE name = 'Clue';
-- return the prices of the game at game shops
SELECT shop, price FROM fun.inventory WHERE game = 'Clue';
```



```
-- set a variable containing the name of the game
SET hivevar:game=Monopoly;
-- return the list price of the game
SELECT list_price FROM fun.games WHERE name = '${hivevar:game}';
-- return the prices of the game at game shops
SELECT shop, price FROM fun.inventory WHERE game = '${hivevar:game}';
```



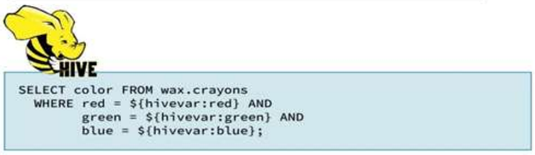
```
SELECT hex FROM wax.crayons WHERE color = '${hivevar:color}';
```



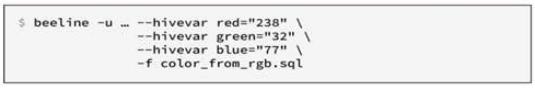
```
$ beeline -u … --hivevar color="Red" -f hex_color.sql
```

```
$ beeline -u … --hivevar color="Orange" -f hex_color.sql
```

```
$ beeline -u … --hivevar color="Yellow" -f hex_color.sql
```



```
SELECT color FROM wax.crayons
  WHERE red = ${hivevar:red} AND
        green = ${hivevar:green} AND
        blue = ${hivevar:blue};
```



```
$ beeline -u … --hivevar red="238" \
               --hivevar green="32" \
               --hivevar blue="77" \
               -f color_from_rgb.sql
```

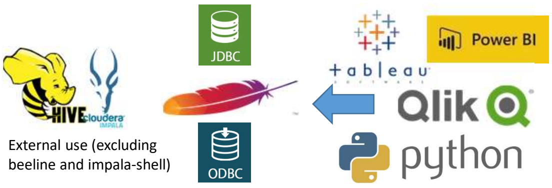
CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

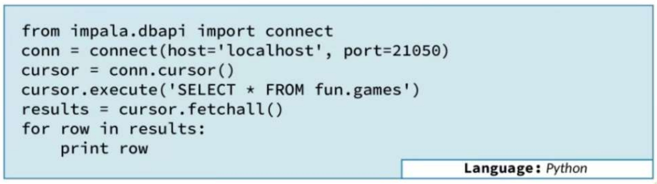
Scripts :

- Clearer vision
- Scheduler of periodic executions
- Do not rewrite the same script
- Usage inside code (python *)
- .Sh extension (chmod 755)
- ./name.sh for execution

*Import subprocess
Subprocess.call([!/script.sh’])



External use (excluding beeline and impala-shell)



```
from impala.dbapi import connect
conn = connect(host='localhost', port=21050)
cursor = conn.cursor()
cursor.execute('SELECT * FROM fun.games')
results = cursor.fetchall()
for row in results:
    print row
```

Language: *Python*

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

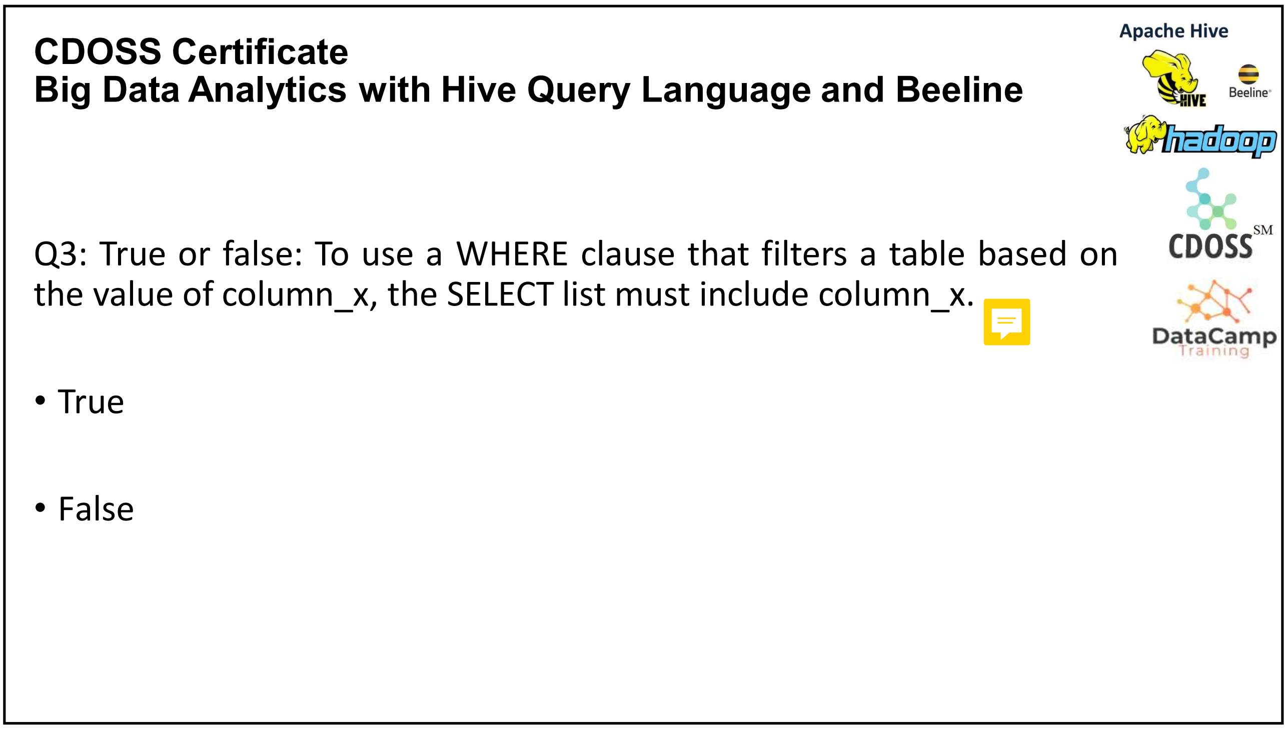
Q1: A table has 100 rows. You use a SELECT statement with a WHERE clause to query the table. Which best describes how many rows the result set must have?

- 100 or fewer
- 100 or more
- More than 100

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q2: For which of these tasks would you need to use a WHERE clause?

- For a table of web logs which show the IP addresses of every visit, removing rows with duplicate IP addresses
- For a table of pets, including their owners and ages, finding the range of values in their ages
- For a table of inventory items, including quantity and price, finding all inventory items priced under $5
- For a table that includes which of many offices each employee works, finding all the employees in the Chicago office

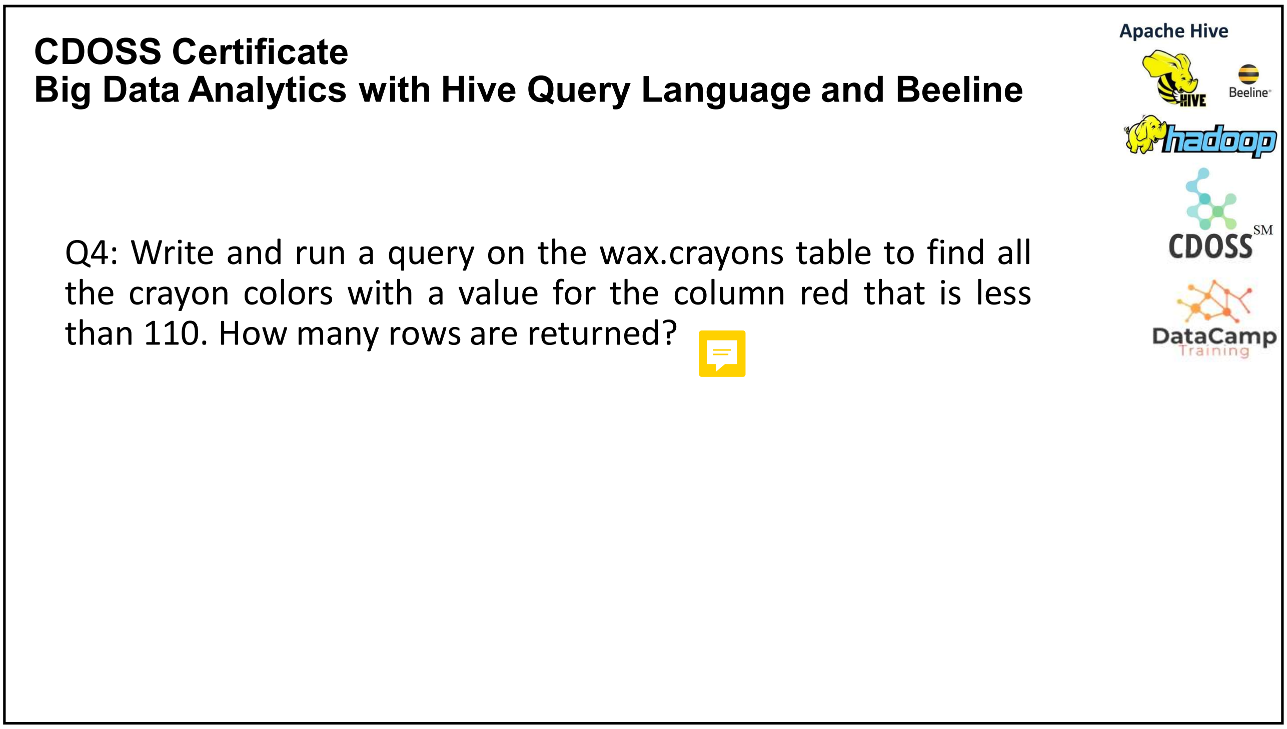


CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive
HIVE
Beeline
hadoop
CDOSS SM
DataCamp
Training

Q3: True or false: To use a WHERE clause that filters a table based on the value of column_x, the SELECT list must include column_x.

- True
- False



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive
HIVE
Beeline
hadoop
CDOSS SM
DataCamp
Training

Q4: Write and run a query on the wax.crayons table to find all the crayon colors with a value for the column red that is less than 110. How many rows are returned?

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q5: The following shows just a few rows from a table for students in a school. (GPA is grade point average, where 4.0 means the student is getting the highest scores possible. Absences is how many days the student has not attended school, and detention is a punishment for bad behavior.)

id	name	age	gpa	absences	detentions
930	Olufunmilayo Ayton	16	4.00	3	2
667	Vincent Michaelson	15	2.53	12	0
907	Asa Quigg	15	3.57	1	0
168	Kiran Patil	17	3.28	0	3

You're asked to find the most dedicated students to represent the school at a state-wide meeting. Which of the following might be appropriate, even though they might give different results? Check all that apply.

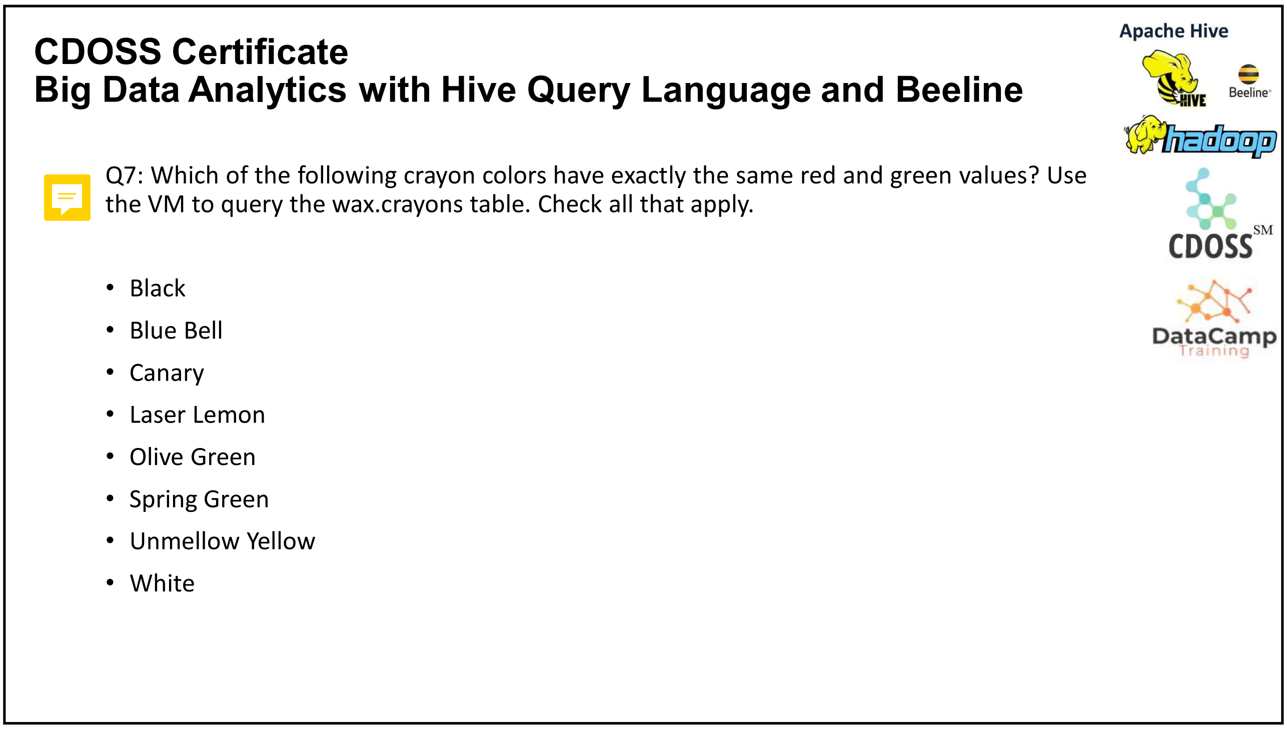
- SELECT name ... WHERE gpa >= 3.5;
- SELECT name ... WHERE detentions = 0;
- SELECT name ... WHERE id < 200;
- SELECT name ... WHERE absences = 0;
- SELECT name ... WHERE absences > detentions;
- SELECT name ... WHERE age = 17;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q6: Which of the following evaluate as true? Check all that apply.

- 8 * -3 != -30 + 5
- 3 >= 1
- 2 * -12 > 6 * -4
- 10 != 10

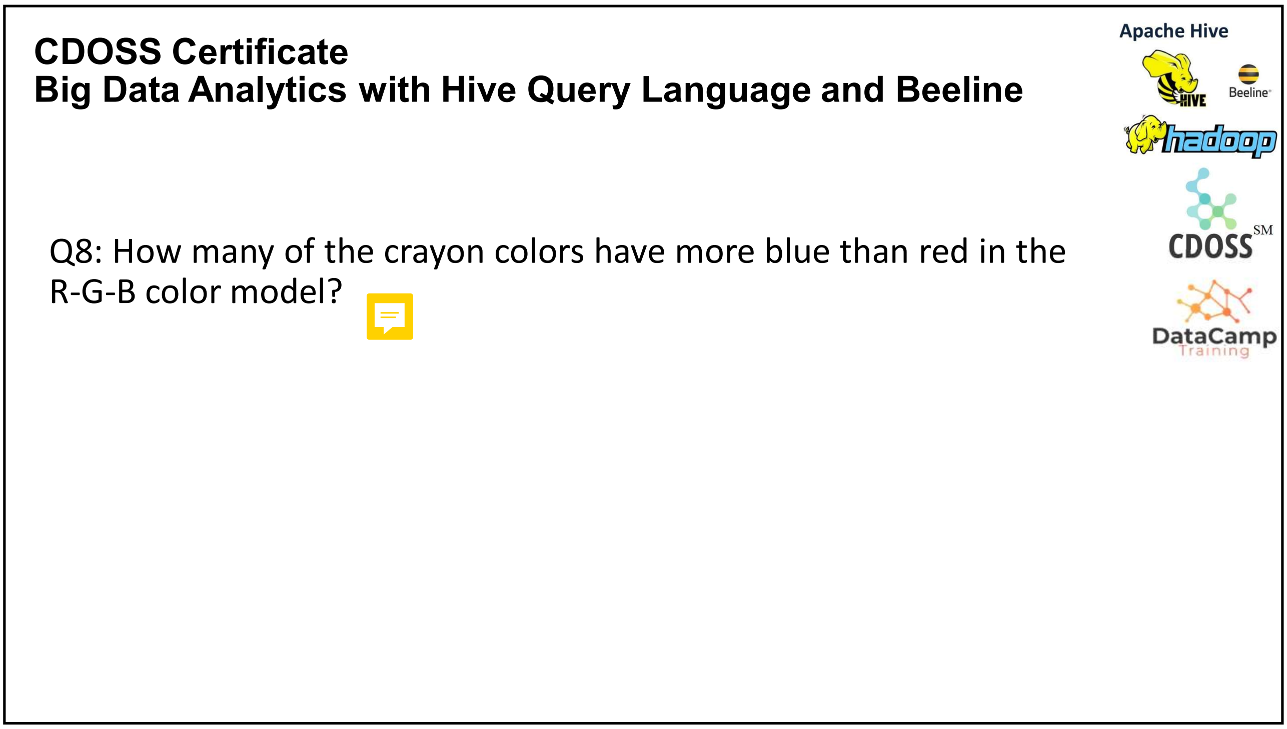


CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q7: Which of the following crayon colors have exactly the same red and green values? Use the VM to query the wax.crayons table. Check all that apply.

- Black
- Blue Bell
- Canary
- Laser Lemon
- Olive Green
- Spring Green
- Unmellow Yellow
- White



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q8: How many of the crayon colors have more blue than red in the R-G-B color model?

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q9: Which of the following crayon colors are dark, that is, the sum of red, green, and blue values will be less than 325? (Not all colors meeting the criteria are listed.) Check all that apply.

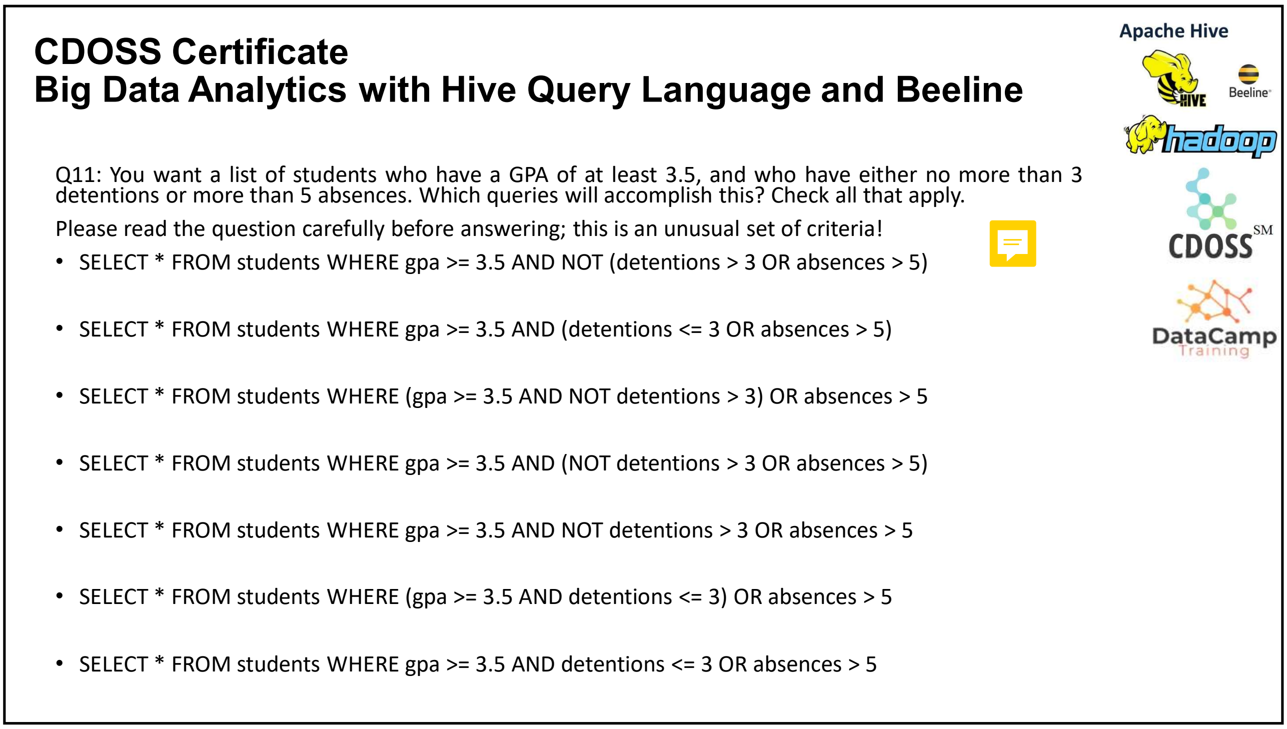
Note: Although it's not needed to answer the question, try writing a query whose results include a column named dark, which is true when the sum is less than 325. The result set for this query should only show rows where dark is true.

- Denim
- Eggplant
- Electric Lime
- Outer Space
- Plum
- Red
- Sepia
- Tropical Rain Forest

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q10: You have a table with integer columns int_x and int_y. Which expressions are valid in SQL? Check all that apply.

- int_x NOT = 2
- int_x OR int_y = 3
- int_x != 2
- int_x = 2 AND int_y = 3
- int_x = 2 & int_y = 3
- NOT int_x = 2

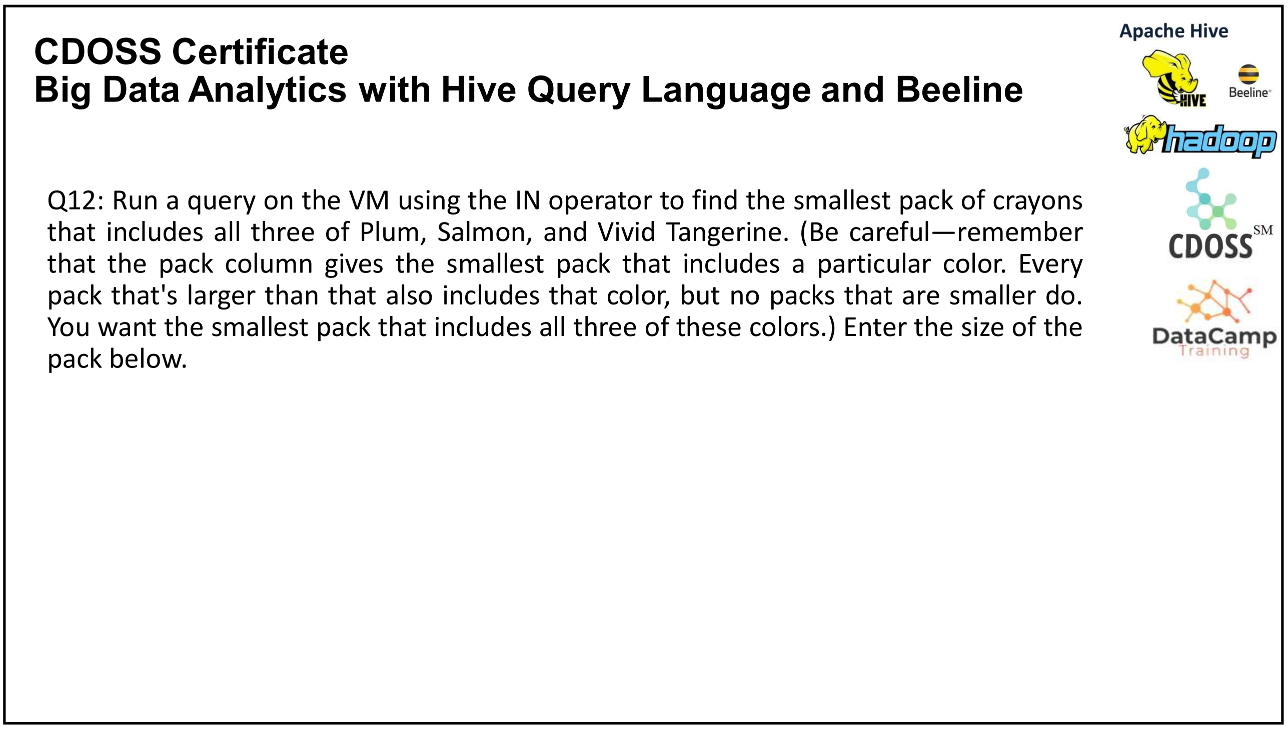


CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q11: You want a list of students who have a GPA of at least 3.5, and who have either no more than 3 detentions or more than 5 absences. Which queries will accomplish this? Check all that apply.

Please read the question carefully before answering; this is an unusual set of criteria!

- SELECT * FROM students WHERE gpa >= 3.5 AND NOT (detentions > 3 OR absences > 5)
- SELECT * FROM students WHERE gpa >= 3.5 AND (detentions <= 3 OR absences > 5)
- SELECT * FROM students WHERE (gpa >= 3.5 AND NOT detentions > 3) OR absences > 5
- SELECT * FROM students WHERE gpa >= 3.5 AND (NOT detentions > 3 OR absences > 5)
- SELECT * FROM students WHERE gpa >= 3.5 AND NOT detentions > 3 OR absences > 5
- SELECT * FROM students WHERE (gpa >= 3.5 AND detentions <= 3) OR absences > 5
- SELECT * FROM students WHERE gpa >= 3.5 AND detentions <= 3 OR absences > 5



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q12: Run a query on the VM using the IN operator to find the smallest pack of crayons that includes all three of Plum, Salmon, and Vivid Tangerine. (Be careful—remember that the pack column gives the smallest pack that includes a particular color. Every pack that's larger than that also includes that color, but no packs that are smaller do. You want the smallest pack that includes all three of these colors.) Enter the size of the pack below.

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q13: Run a query on the VM to find which of the following crayon colors has a red value between 75 and 125 and a blue value between 125 and 175. Check all that apply.

- Forest Green
- Manatee
- Royal Purple
- Screamin' Green
- Shadow

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

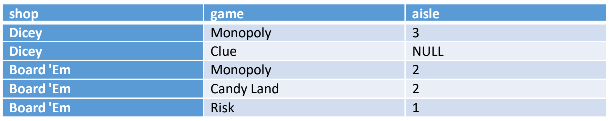
Q14: Write a query to return all rows for a flight in the flights table with the following information: It departed on January 15, 2009, the carrier is capital letters US, the flight number is 1549, and the origin airport is capital letters LGA. Which column or columns in this row have NULL values? Check all that apply.

- air_time
- arr_delay
- arr_time
- dep_time

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q15: How many different games in the fun.inventory table are located in Aisle 3 of the Dicey shop? Check the best answer.



shop	game	aisle
Dicey	Monopoly	3
Dicey	Clue	NULL
Board 'Em	Monopoly	2
Board 'Em	Candy Land	2
Board 'Em	Risk	1

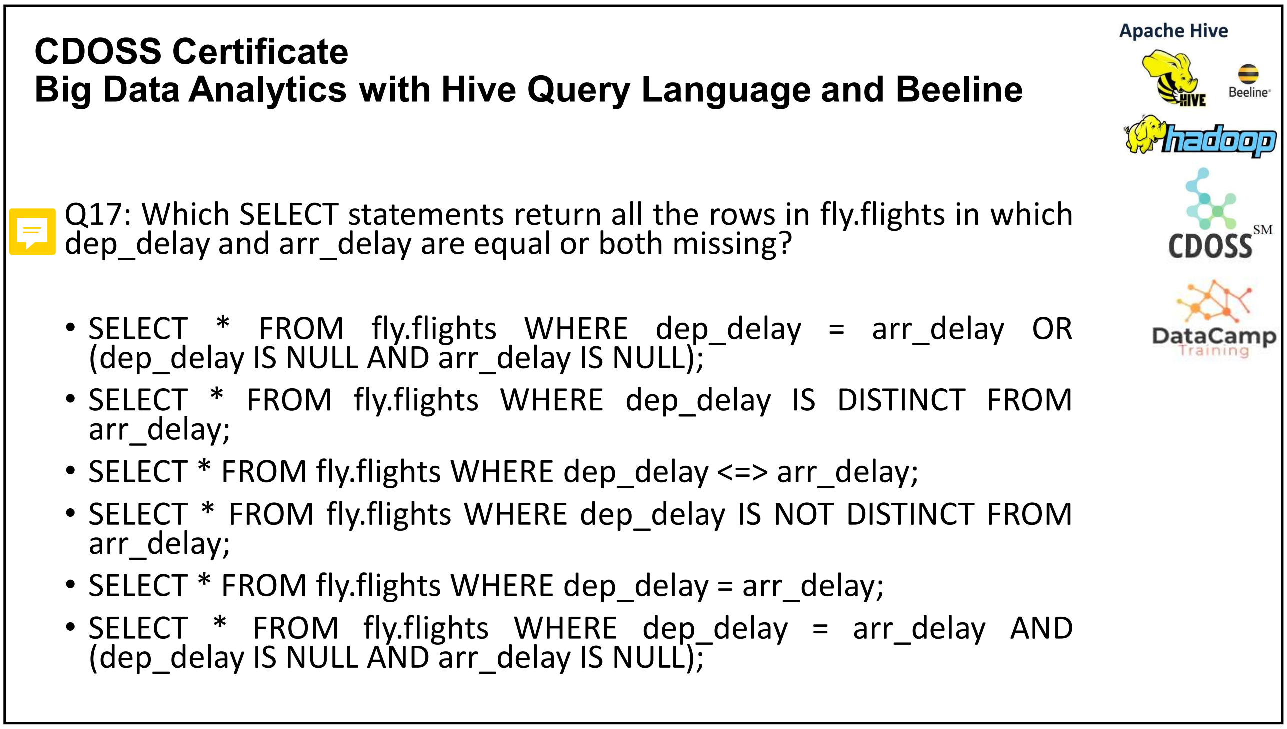
- None
- One
- At least one
- At most one
- More than one

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q16: On the VM, write and run a SELECT statement that queries the fly.flights table and returns all the rows representing flights on January 15, 2009 that have non-missing departure time (dep_time) and missing arrival time (arr_time). You'll need to use both IS NULL and IS NOT NULL to do this.

How many rows are returned?

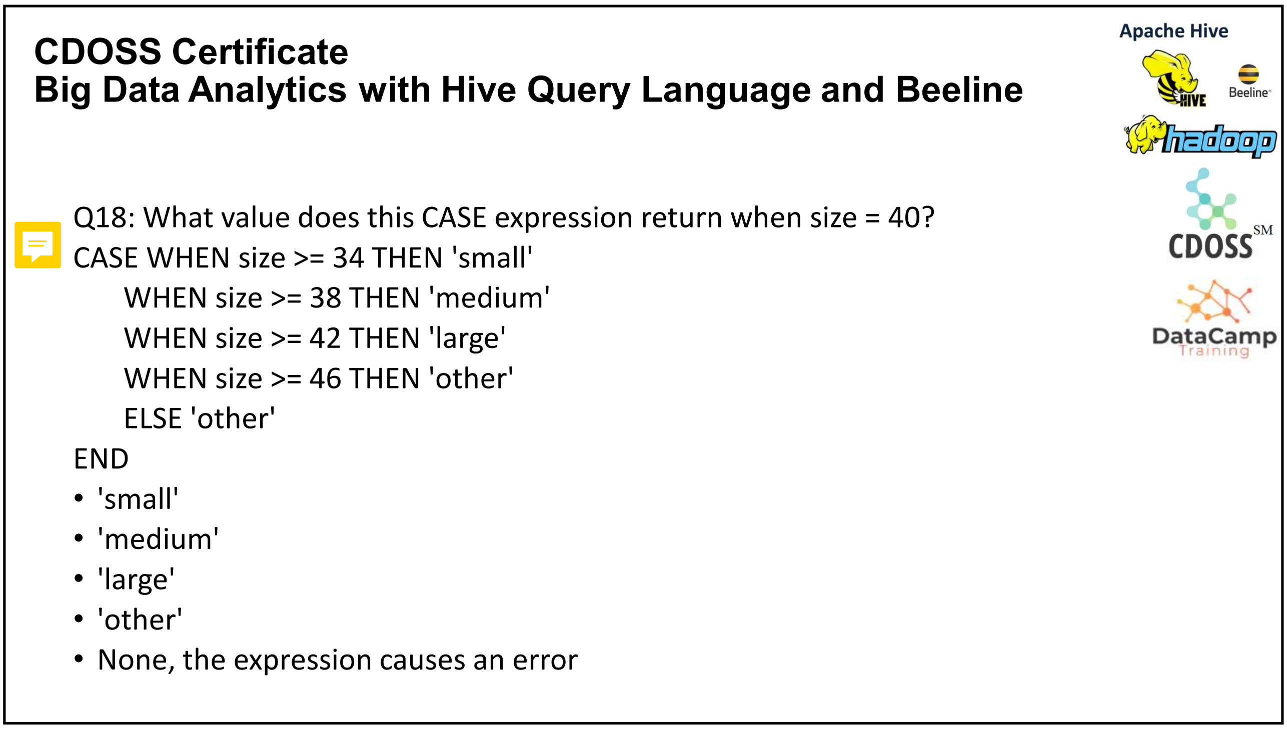


CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q17: Which SELECT statements return all the rows in fly.flights in which dep_delay and arr_delay are equal or both missing?

- SELECT * FROM fly.flights WHERE dep_delay = arr_delay OR (dep_delay IS NULL AND arr_delay IS NULL);
- SELECT * FROM fly.flights WHERE dep_delay IS DISTINCT FROM arr_delay;
- SELECT * FROM fly.flights WHERE dep_delay <=> arr_delay;
- SELECT * FROM fly.flights WHERE dep_delay IS NOT DISTINCT FROM arr_delay;
- SELECT * FROM fly.flights WHERE dep_delay = arr_delay;
- SELECT * FROM fly.flights WHERE dep_delay = arr_delay AND (dep_delay IS NULL AND arr_delay IS NULL);



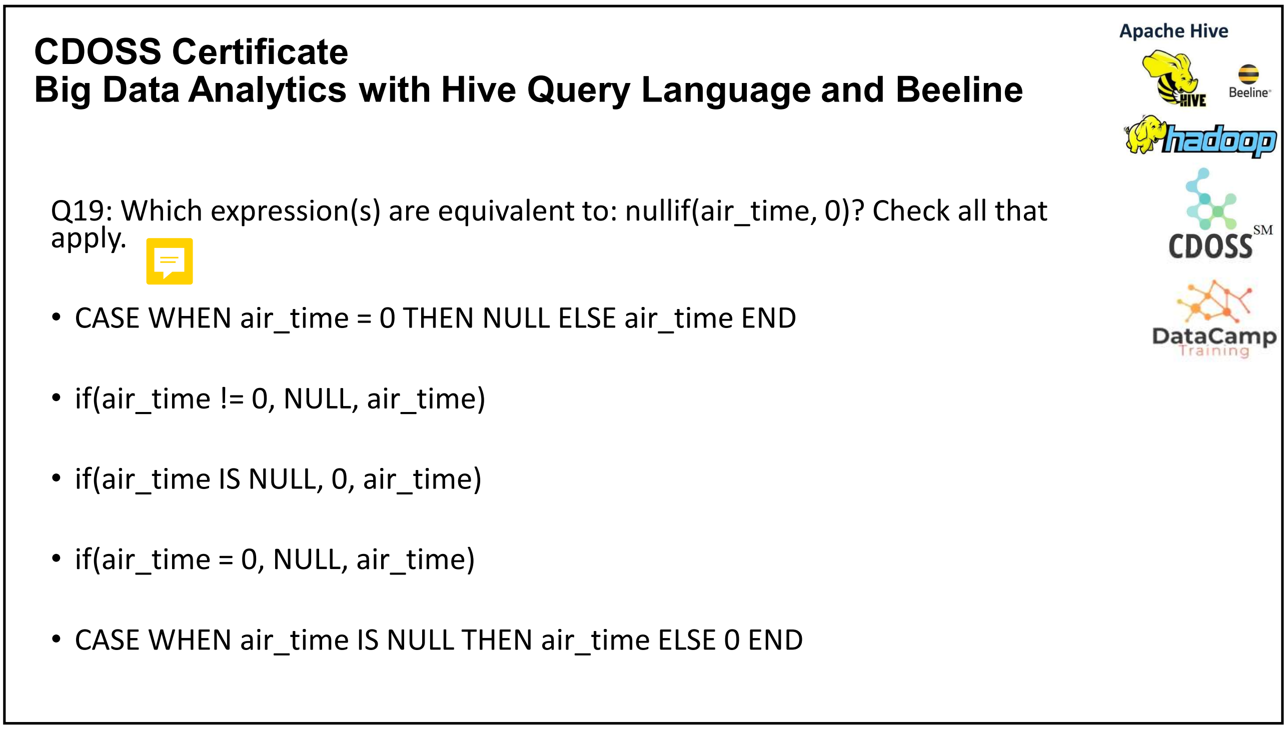
CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q18: What value does this CASE expression return when size = 40?

```
CASE WHEN size >= 34 THEN 'small'
     WHEN size >= 38 THEN 'medium'
     WHEN size >= 42 THEN 'large'
     WHEN size >= 46 THEN 'other'
     ELSE 'other'
END
```

- 'small'
- 'medium'
- 'large'
- 'other'
- None, the expression causes an error

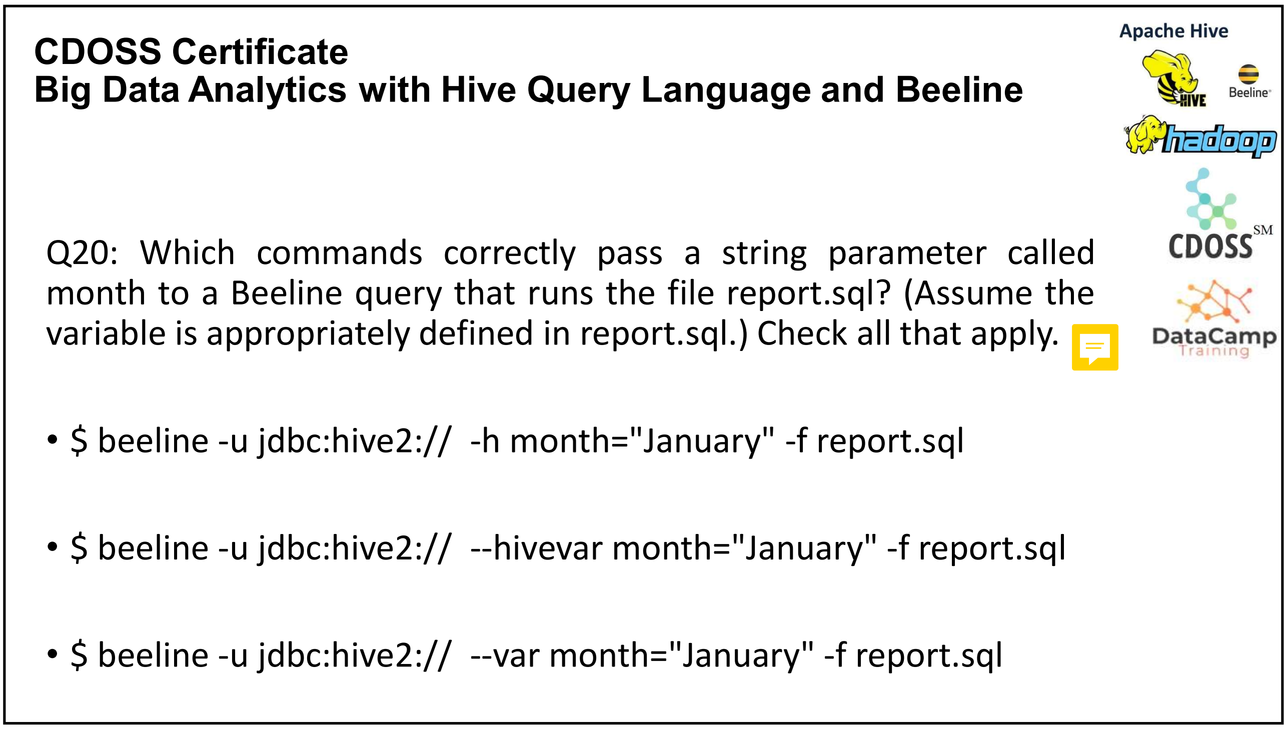


CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive
Beeline
hadoop
CDOSS
DataCamp
Training

Q19: Which expression(s) are equivalent to: nullif(air_time, 0)? Check all that apply.

- CASE WHEN air_time = 0 THEN NULL ELSE air_time END
- if(air_time != 0, NULL, air_time)
- if(air_time IS NULL, 0, air_time)
- if(air_time = 0, NULL, air_time)
- CASE WHEN air_time IS NULL THEN air_time ELSE 0 END



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive
Beeline
hadoop
CDOSS
DataCamp
Training

Q20: Which commands correctly pass a string parameter called month to a Beeline query that runs the file report.sql? (Assume the variable is appropriately defined in report.sql.) Check all that apply.

- $ beeline -u jdbc:hive2:// -h month="January" -f report.sql
- $ beeline -u jdbc:hive2:// --hivevar month="January" -f report.sql
- $ beeline -u jdbc:hive2:// --var month="January" -f report.sql

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q21: Suppose that, as you are working, you need to run a bash script query_script.sh with a SQL query in it. (That is, you want to run it now, not schedule it for later.) You have never run this script before. Which of the following is necessary to run the script? Check all that apply. (Note that the order provided might not match the order in which you need to proceed.)

- Run the script from the command line using $ bash query_script.sh (assuming it is in your current directory)
- Run the script from Beeline shell using BASH query_script.sh; (assuming it is in your current directory)
- Run the script from the command line using $ ./query_script.sh (assuming it is in your current directory)
- Give permission to the script using chmod
- Run the script from Beeline shell using RUN query_script.sh; (assuming it is in your current directory)
- Use the root or superuser privileges when issuing the run command, so the script has permissions to run

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

- Aggregation

Aggregation functions
Example : MAX, MIN, AVG, SUM…

Aggregation functions and scalar functions (example: ABS, ROUND ...) never together

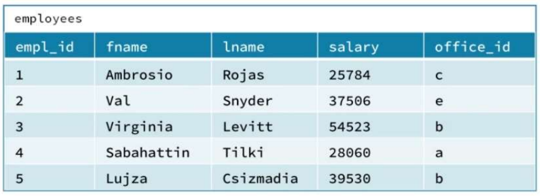
- AVG(list_price), ABS(list_price) ✗
- SELECT salary-AVG(salary) ✗
- SELECT first_name, SUM(salary) ✗
- Aggregation functions in WHERE ✗

LEAST # MIN   GREATEST # MAX
LEAST (15,20,10)=10
→ these are not aggregation functions

GROUP BY

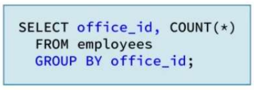
WHERE preced GROUP BY
→ Row filtering before grouping

How many employees are there *in each office*?

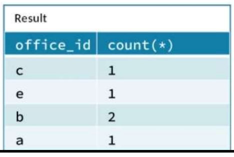


employees

empl_id	fname	lname	salary	office_id
1	Ambrosio	Rojas	25784	c
2	Val	Snyder	37506	e
3	Virginia	Levitt	54523	b
4	Sabahattin	Tilki	28060	a
5	Lujza	Csizmadia	39530	b

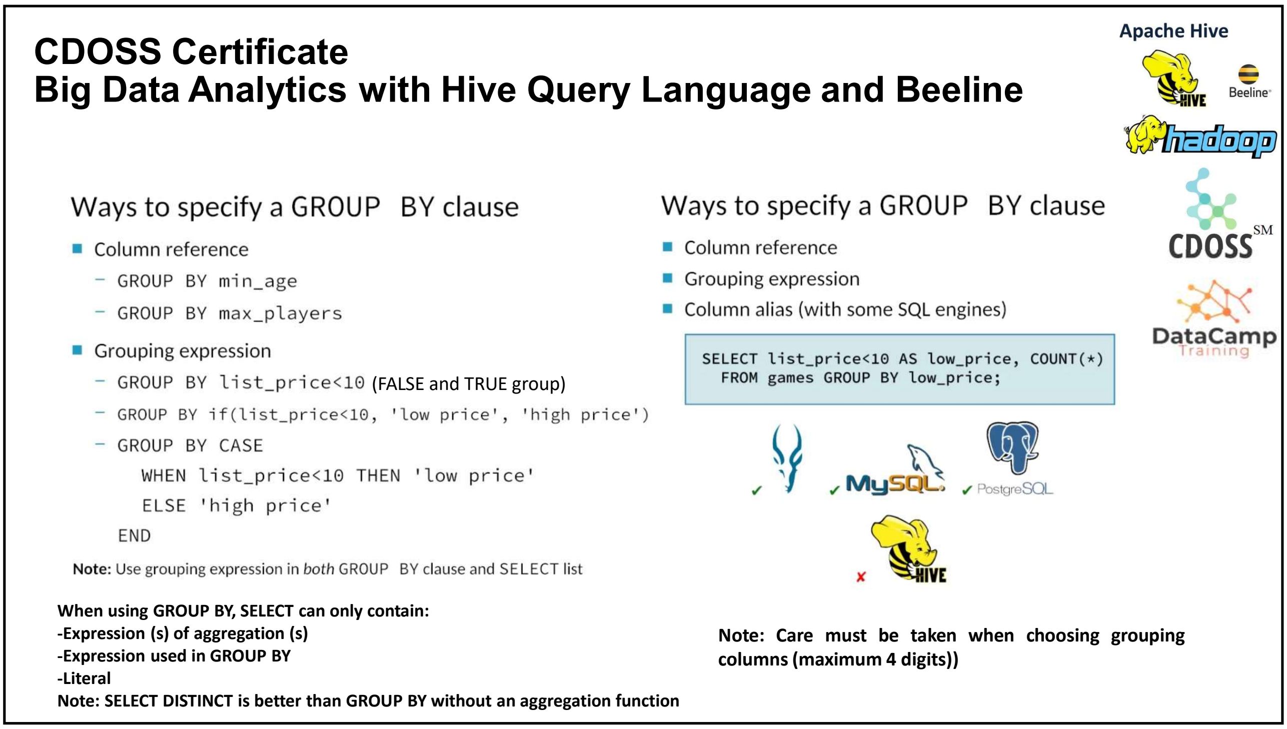


```
SELECT office_id, COUNT(*)
  FROM employees
  GROUP BY office_id;
```



Result

office_id	count(*)
c	1
e	1
b	2
a	1



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

HIVE Beeline hadoop CDOSS℠ DataCamp Training

Ways to specify a GROUP BY clause

- Column reference
  - `GROUP BY min_age`
  - `GROUP BY max_players`
- Grouping expression
  - `GROUP BY list_price<10` (FALSE and TRUE group)
  - `GROUP BY if(list_price<10, 'low price', 'high price')`
  - GROUP BY CASE

```
GROUP BY CASE
  WHEN list_price<10 THEN 'low price'
  ELSE 'high price'
END
```

Note: Use grouping expression in *both* GROUP BY clause and SELECT list

When using GROUP BY, SELECT can only contain:
-Expression (s) of aggregation (s)
-Expression used in GROUP BY
-Literal
Note: SELECT DISTINCT is better than GROUP BY without an aggregation function

Ways to specify a GROUP BY clause

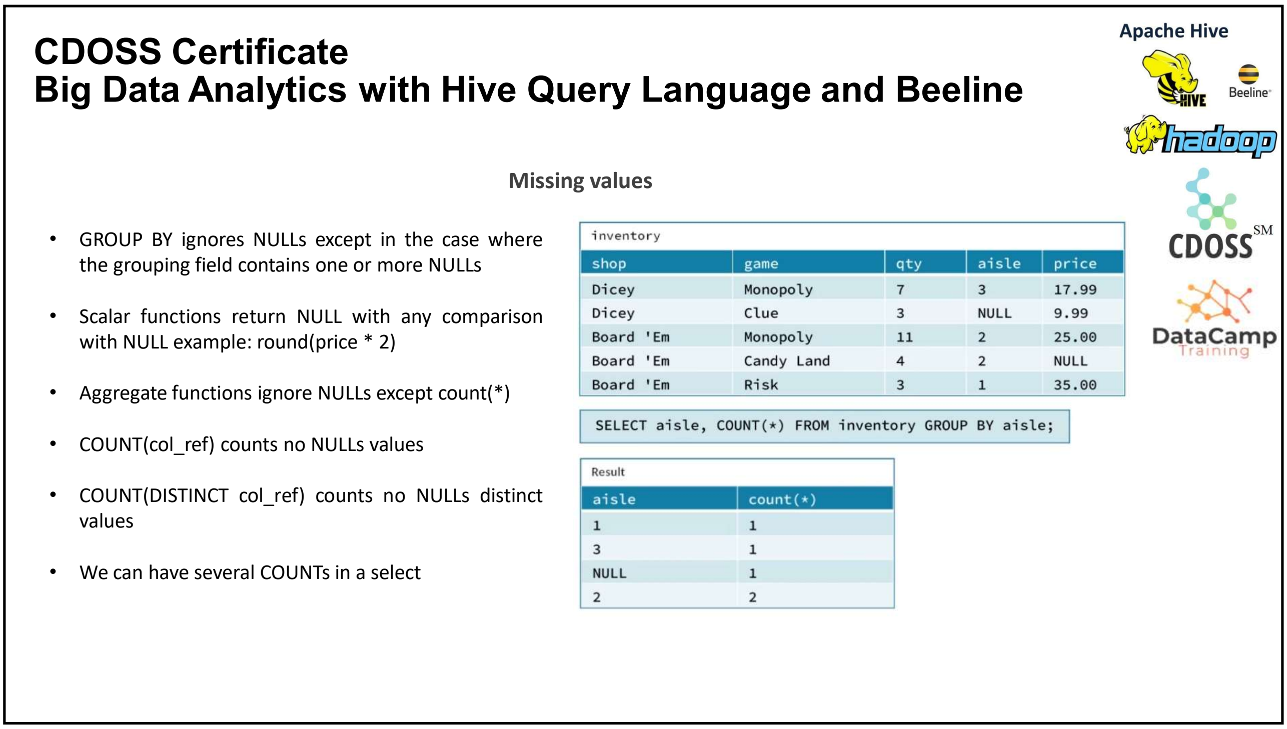
- Column reference
- Grouping expression
- Column alias (with some SQL engines)

```
SELECT list_price<10 AS low_price, COUNT(*)
  FROM games GROUP BY low_price;
```

✓ ✓ MySQL ✓ PostgreSQL

✗ HIVE

Note: Care must be taken when choosing grouping columns (maximum 4 digits))



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

HIVE Beeline hadoop CDOSS℠ DataCamp Training

Missing values

- GROUP BY ignores NULLs except in the case where the grouping field contains one or more NULLs
- Scalar functions return NULL with any comparison with NULL example: round(price * 2)
- Aggregate functions ignore NULLs except count(*)
- COUNT(col_ref) counts no NULLs values
- COUNT(DISTINCT col_ref) counts no NULLs distinct values
- We can have several COUNTs in a select

inventory

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00

```
SELECT aisle, COUNT(*) FROM inventory GROUP BY aisle;
```

Result

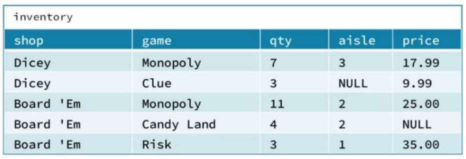
aisle	count(*)
1	1
3	1
NULL	1
2	2

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

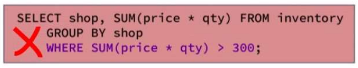
Beeline

HAVING : Group filtering



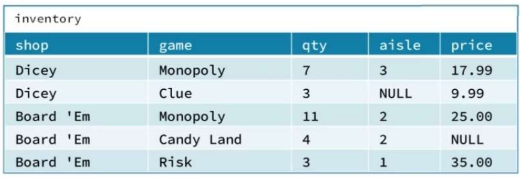
inventory

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00



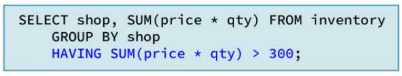
```
SELECT shop, SUM(price * qty) FROM inventory
  GROUP BY shop
  WHERE SUM(price * qty) > 300;
```

→ WHERE is used before GROUP BY so does not filter groups

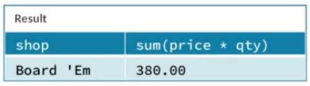


inventory

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00



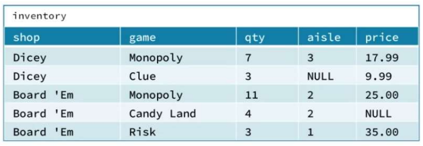
```
SELECT shop, SUM(price * qty) FROM inventory
    GROUP BY shop
    HAVING SUM(price * qty) > 300;
```



Result

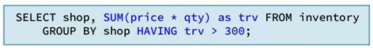
shop	sum(price * qty)
Board 'Em	380.00

Alias



inventory

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00



```
SELECT shop, SUM(price * qty) as trv FROM inventory
    GROUP BY shop HAVING trv > 300;
```

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Beeline

- Best practice : use the pushdown for Big Data (with aggregations, groupings and filters) before using them with BI tools to avoid crashes and exorbitant transfer costs

Power BI

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q1: Below is a table with four rows. What is the value of SUM(items) for this table?

order_id	items	total
829	3	38.92
220	5	107.06
1043	2	19.98
762	1	20.49

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q2: What is the average list price of the games in the fun.games table in US dollars? Use the virtual machine (VM) to calculate this.

The query should be similar to

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q3: If 1 US dollar is equivalent to 66.75 Indian rupees, what is the average list price of the games in the fun.games table in Indian rupees, rounded to two places after the decimal? Use the VM to calculate this.

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q4: You could also have rounded the converted amounts and then found the average; in this case, both calculations return the same value.

Which of the following statements are valid? Check all that apply.

- SELECT SUM(1.06 * price) FROM fun.inventory;
- SELECT SUM(qty * price) FROM fun.inventory;
- SELECT 1.06 * SUM(price) FROM fun.inventory;
- SELECT qty * SUM(price) FROM fun.inventory;
- SELECT game, SUM(price) FROM fun.inventory;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

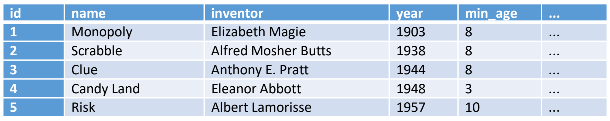
Apache Hive

Q5: The flights dataset includes the distance (in miles) and time (in minutes) in the air for the included flights. Write and run a query to find the average air speed, in miles per hour, of only those flights that were in the air for longer than 60 minutes. Report to the nearest mile per hour. (Hints: Speed is distance divided by time. Remember that the time is in minutes, and this problem asks for speed in miles per hour. Your answer should be an integer.)

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q6: Here is a portion of the fun.games table. The table has 8 columns, but not all are shown (for space considerations).



id	name	inventor	year	min_age	...
1	Monopoly	Elizabeth Magie	1903	8	...
2	Scrabble	Alfred Mosher Butts	1938	8	...
3	Clue	Anthony E. Pratt	1944	8	...
4	Candy Land	Eleanor Abbott	1948	3	...
5	Risk	Albert Lamorisse	1957	10	...

How many columns and rows does the result of this query have? SELECT min_age, COUNT(*) FROM fun.games GROUP BY min_age; Please attempt to answer this question without actually running the query.

- 2 columns, 1 row
- 2 columns, 3 rows
- 2 columns, 5 rows
- 8 columns, 1 row
- 8 columns, 3 rows
- 8 columns, 5 rows
- 10 columns, 1 row
- 10 columns, 3 rows
- 10 columns, 5 rows

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q7: Here is a portion of the fun.games table. The table has 8 columns, but not all are shown (for space considerations).

id	name	inventor	year	min_age	...
1	Monopoly	Elizabeth Magie	1903	8	...
2	Scrabble	Alfred Mosher Butts	1938	8	...
3	Clue	Anthony E. Pratt	1944	8	...
4	Candy Land	Eleanor Abbott	1948	3	...
5	Risk	Albert Lamorisse	1957	10	...

Which values occur in the second column of the result of the following query? Check all that apply. SELECT min_age, COUNT(*) FROM fun.games GROUP BY min_age;

- 1
- 2
- 3
- 8
- 10

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q8: Run the following query on the VM using hive. Then use the result set to answer the following question:

```
SELECT min_age, COUNT(*)
   FROM fun.games
   WHERE list_price > 10
   GROUP BY min_age;
```

How many games with a list price greater than $10 are suitable for players as young as 3?

- Unknown
- 0
- 1
- 2
- 3

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

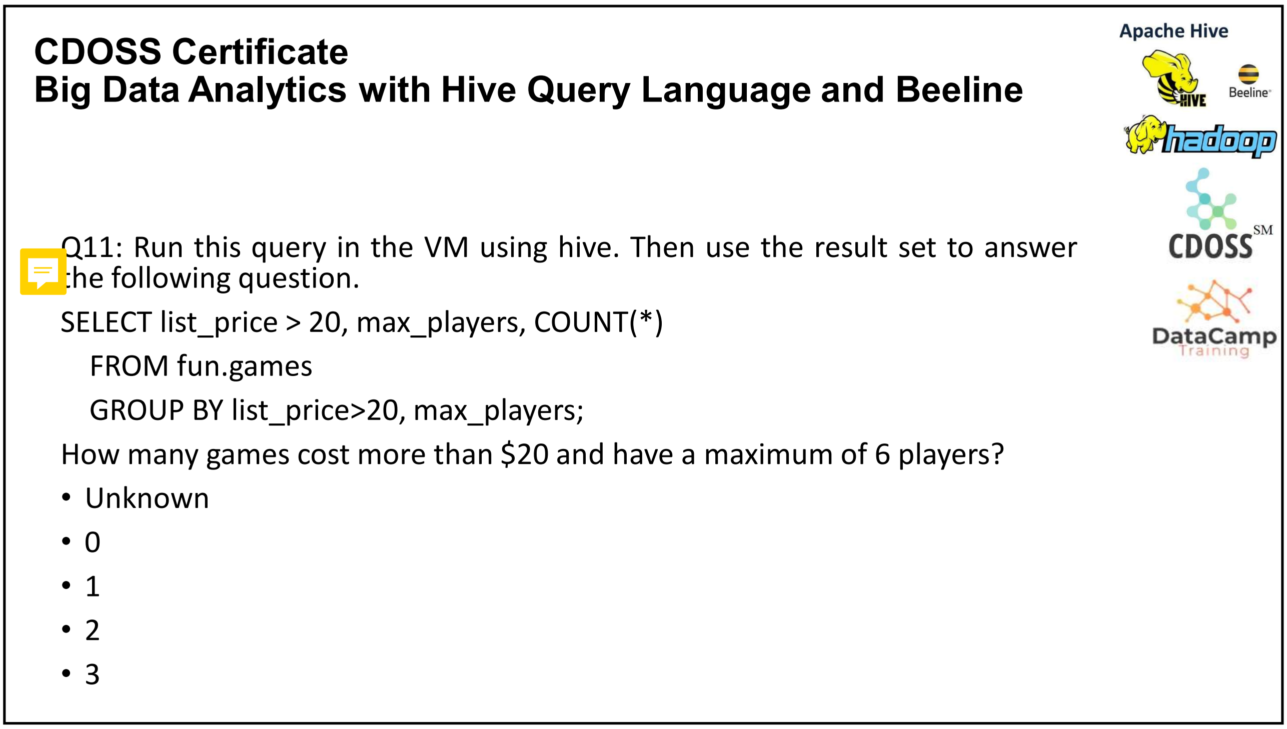
Q9: Write and run a query on the fly.planes table that would answer the question, "What is the average number of seats for each type of aircraft in the table?" Then use the results to enter the average number of seats for the blimps/dirigibles in the table.

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q10: Which of these expressions runs without error in Hive? Check all that apply. (If needed, check your answers by attempting to run these queries in Hive.)

- SELECT list_price < 10, COUNT(*) FROM fun.games GROUP BY list_price < 10;
- SELECT list_price < 10 AS low_price, COUNT(*) FROM fun.games GROUP BY low_price;
- SELECT list_price < 10 AS low_price, COUNT(*) FROM fun.games GROUP BY list_price < 10;
- SELECT low_price, COUNT(*) FROM fun.games GROUP BY list_price < 10 AS low_price;



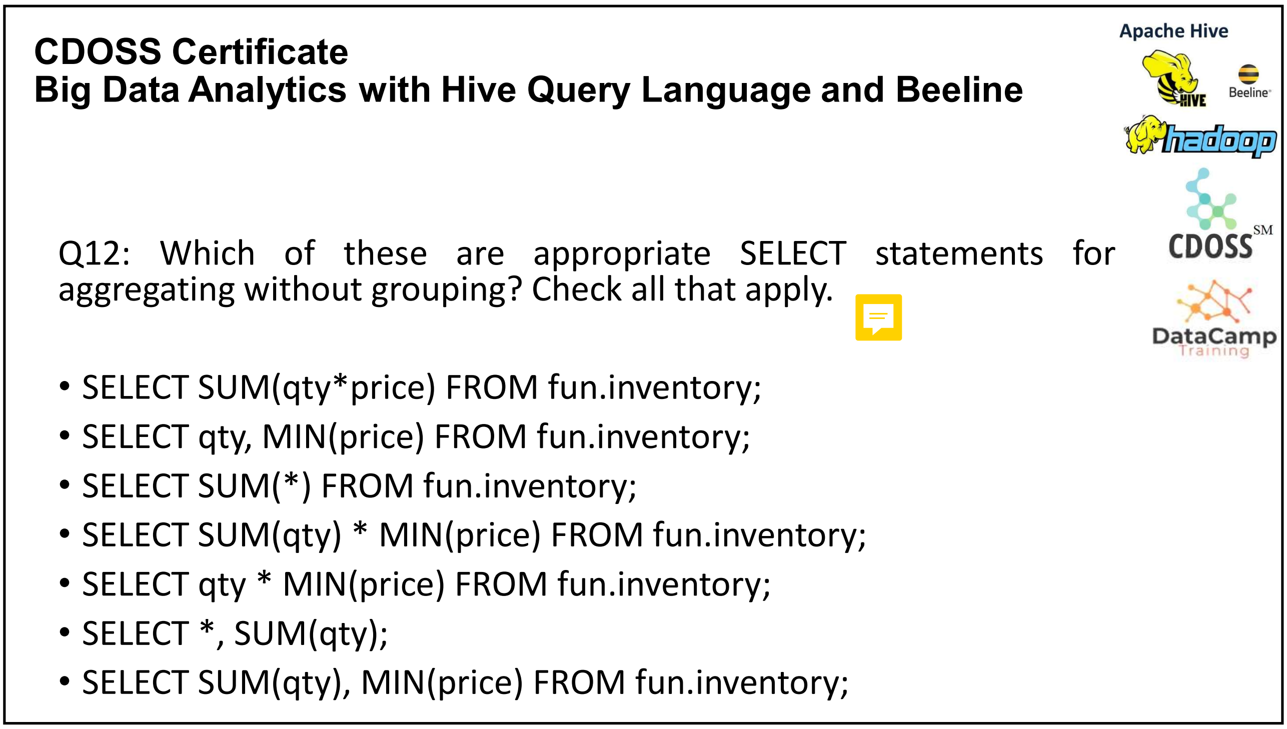
CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q11: Run this query in the VM using hive. Then use the result set to answer the following question.

```
SELECT list_price > 20, max_players, COUNT(*)
   FROM fun.games
   GROUP BY list_price>20, max_players;
```

How many games cost more than $20 and have a maximum of 6 players?

- Unknown
- 0
- 1
- 2
- 3



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q12: Which of these are appropriate SELECT statements for aggregating without grouping? Check all that apply.

- SELECT SUM(qty*price) FROM fun.inventory;
- SELECT qty, MIN(price) FROM fun.inventory;
- SELECT SUM(*) FROM fun.inventory;
- SELECT SUM(qty) * MIN(price) FROM fun.inventory;
- SELECT qty * MIN(price) FROM fun.inventory;
- SELECT *, SUM(qty);
- SELECT SUM(qty), MIN(price) FROM fun.inventory;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q13: Which of these are appropriate SELECT statements for grouping without aggregation? Check all that apply.

- SELECT * FROM fun.inventory GROUP BY qty;
- SELECT qty * 2 FROM fun.inventory GROUP BY qty;
- SELECT qty FROM fun.inventory GROUP BY qty;
- SELECT name FROM fun.inventory GROUP BY qty;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q14: Here is the fun.inventory table:

shop	game	qty	aisle	price
Dicey	Monopoly	7	3	17.99
Dicey	Clue	3	NULL	9.99
Board 'Em	Monopoly	11	2	25.00
Board 'Em	Candy Land	4	2	NULL
Board 'Em	Risk	3	1	35.00

Without running this query on the VM, predict what value it will return:

SELECT MIN(price) FROM fun.inventory;

- NULL
- 0
- 9.99
- 21.99
- 34.00
- 35.00

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q15: The query SELECT MIN(price) FROM fun.inventory; gave one row in the results, with only one column. The value was 9.99.

Choose which of the following statements is most accurate and informative.

- The lowest price of a game in the inventory table is $9.99.
- The lowest known price of a game in the inventory table is $9.99.
- The lowest price of a game in the inventory table is unknown.

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q16: Write and run a query using hive to find the average air speed (distance divided by air_time) of all flights in the fly.flights table, in miles per hour. Choose the answer below that is most accurate and informative.

- Infinity mi/hr
- About 7 mi/hr
- Impossible to determine
- About 402 mi/hr

The nullif function is needed to prevent division by 0. The query should be similar to SELECT AVG(distance/(nullif(air_time,0)/60)) FROM fly.flights;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q17: Which statement will return the same result as this one?
SELECT AVG(price) AS avg_price FROM fun.inventory;

Try to choose the correct answer without running these SELECT statements. If you are uncertain, check your answer by running it in hive on the VM.

- SELECT SUM(price) / COUNT(price) AS avg_price FROM fun.inventory;
- SELECT SUM(price) / SUM(1) AS avg_price FROM fun.inventory;
- SELECT SUM(price) / COUNT(*) AS avg_price FROM fun.inventory;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q18: Use hive in the VM to find how many unique non-NULL combinations of year, month, and day exist in the fly.flights table.

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q19: Which SELECT statements will return the same result as

SELECT COUNT(tz) AS time_zones FROM fly.airports;

Check all that apply. Try to choose the correct answers without running these SELECT statements. If you are uncertain, check your answers by running them on the VM.

- SELECT COUNT(DISTINCT tz) AS time_zones FROM fly.airports;
- SELECT COUNT(*) AS time_zones FROM fly.airports;
- SELECT COUNT(*) AS time_zones FROM fly.airports WHERE tz IS NOT NULL;
- SELECT COUNT(ALL tz) AS time_zones FROM fly.airports;
- SELECT COUNT(*) AS time_zones FROM fly.airports WHERE tz IS NULL;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

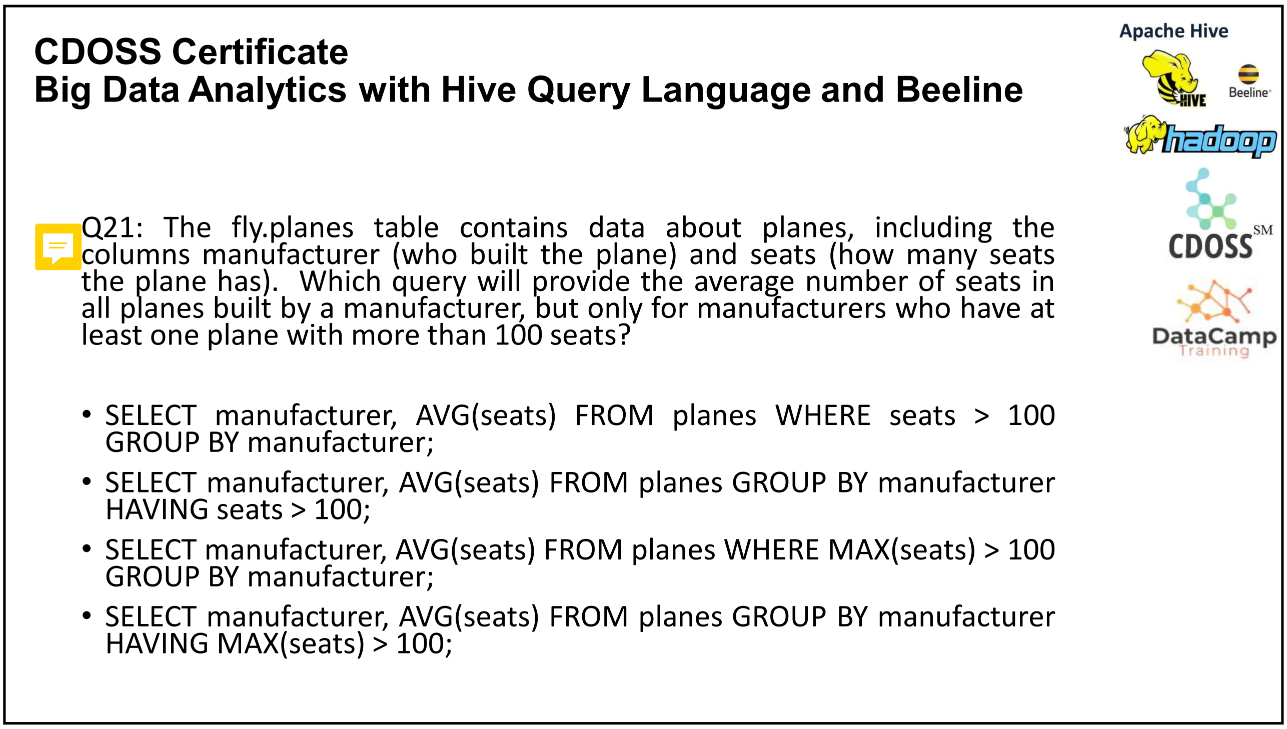
Apache Hive

Q20: The fly.flights table includes the column flight, which gives a flight number for each flight in the table. It also includes the column carrier, which gives the airline for each flight. This query gives the number of carriers that use each particular flight number:

SELECT flight, COUNT(DISTINCT carrier) FROM flights GROUP BY flight;

Which of the following is the best response to whether this is a good choice for grouping? (If you are unsure, you might inspect the data in the VM, looking for maximum and minimum values for the column, or counting the number of distinct values.)

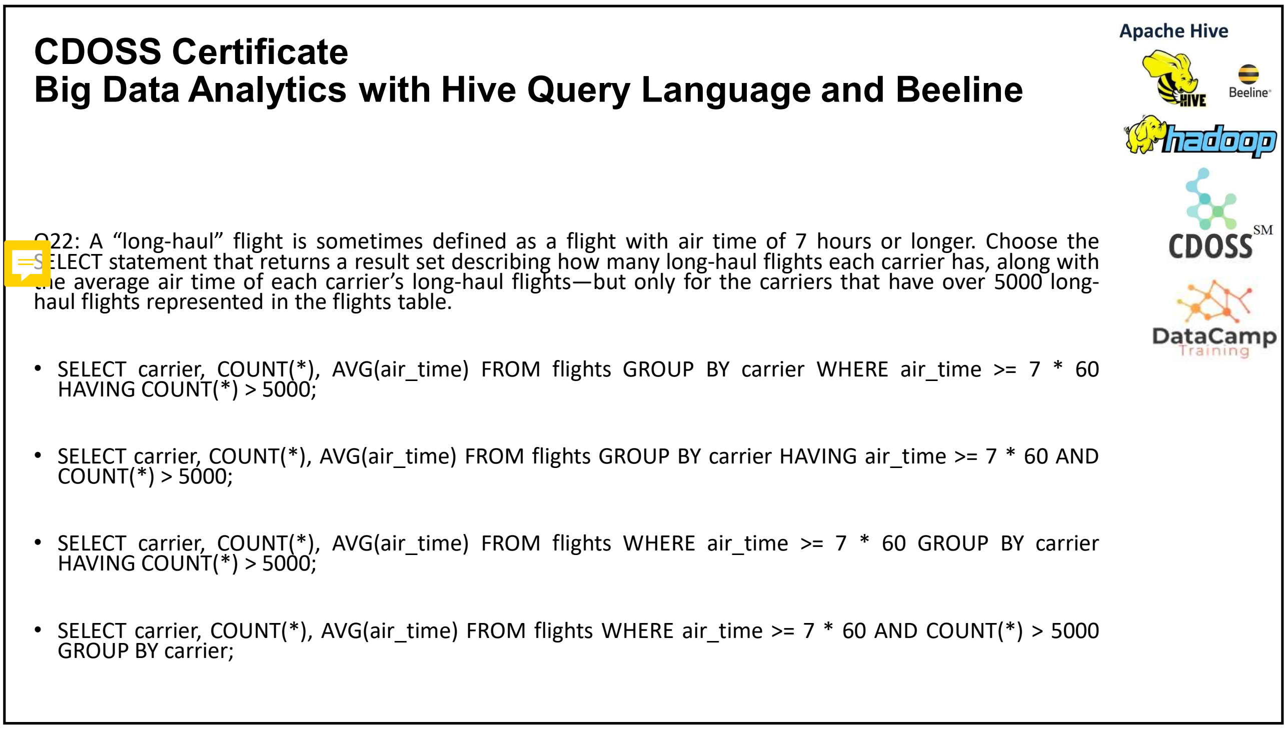
- It's not a good choice, because there could be any number of distinct values.
- It's a good choice because there would only be a few distinct values.
- It's a reasonable choice, because while there might be several thousand values for flight numbers, for big data this is not an unreasonable number of rows.



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q21: The fly.planes table contains data about planes, including the columns manufacturer (who built the plane) and seats (how many seats the plane has). Which query will provide the average number of seats in all planes built by a manufacturer, but only for manufacturers who have at least one plane with more than 100 seats?

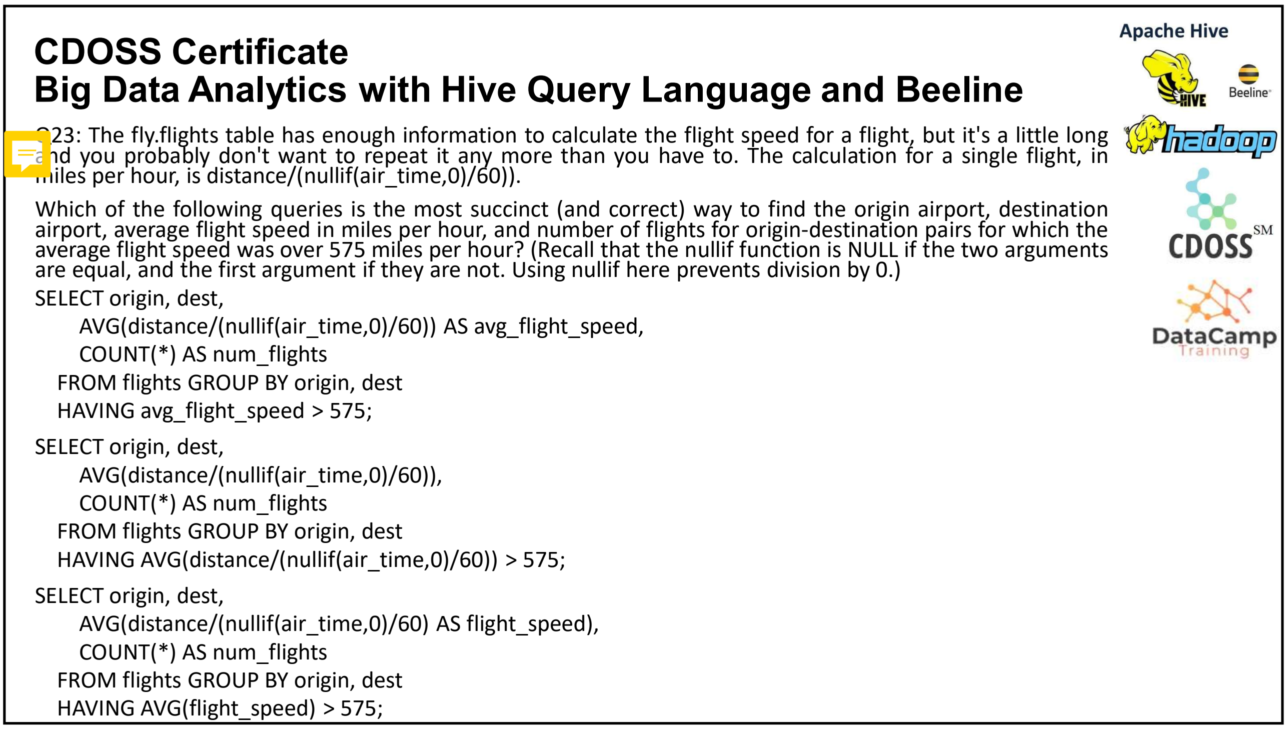
- SELECT manufacturer, AVG(seats) FROM planes WHERE seats > 100 GROUP BY manufacturer;
- SELECT manufacturer, AVG(seats) FROM planes GROUP BY manufacturer HAVING seats > 100;
- SELECT manufacturer, AVG(seats) FROM planes WHERE MAX(seats) > 100 GROUP BY manufacturer;
- SELECT manufacturer, AVG(seats) FROM planes GROUP BY manufacturer HAVING MAX(seats) > 100;



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q22: A “long-haul” flight is sometimes defined as a flight with air time of 7 hours or longer. Choose the SELECT statement that returns a result set describing how many long-haul flights each carrier has, along with the average air time of each carrier’s long-haul flights—but only for the carriers that have over 5000 long-haul flights represented in the flights table.

- SELECT carrier, COUNT(*), AVG(air_time) FROM flights GROUP BY carrier WHERE air_time >= 7 * 60 HAVING COUNT(*) > 5000;
- SELECT carrier, COUNT(*), AVG(air_time) FROM flights GROUP BY carrier HAVING air_time >= 7 * 60 AND COUNT(*) > 5000;
- SELECT carrier, COUNT(*), AVG(air_time) FROM flights WHERE air_time >= 7 * 60 GROUP BY carrier HAVING COUNT(*) > 5000;
- SELECT carrier, COUNT(*), AVG(air_time) FROM flights WHERE air_time >= 7 * 60 AND COUNT(*) > 5000 GROUP BY carrier;



Apache Hive

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

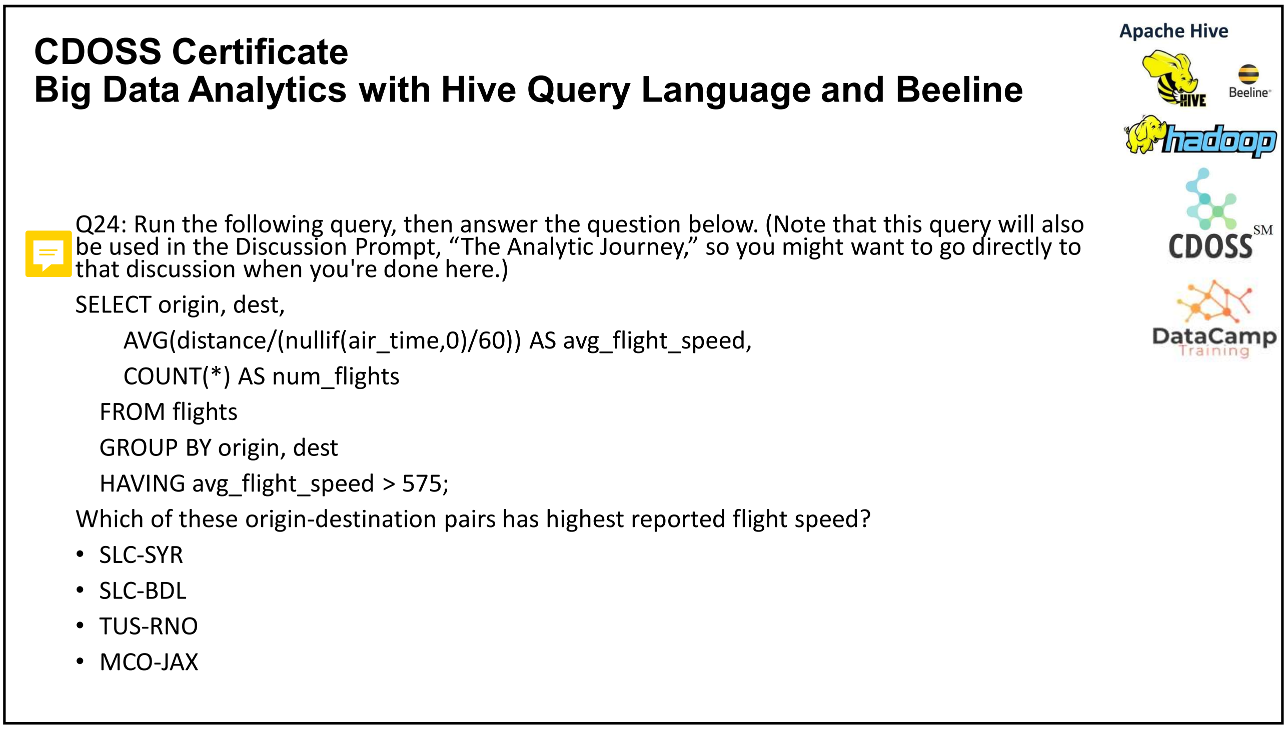
Q23: The fly.flights table has enough information to calculate the flight speed for a flight, but it's a little long and you probably don't want to repeat it any more than you have to. The calculation for a single flight, in miles per hour, is distance/(nullif(air_time,0)/60)).

Which of the following queries is the most succinct (and correct) way to find the origin airport, destination airport, average flight speed in miles per hour, and number of flights for origin-destination pairs for which the average flight speed was over 575 miles per hour? (Recall that the nullif function is NULL if the two arguments are equal, and the first argument if they are not. Using nullif here prevents division by 0.)

```
SELECT origin, dest,
      AVG(distance/(nullif(air_time,0)/60)) AS avg_flight_speed,
      COUNT(*) AS num_flights
   FROM flights GROUP BY origin, dest
   HAVING avg_flight_speed > 575;
```

```
SELECT origin, dest,
      AVG(distance/(nullif(air_time,0)/60)),
      COUNT(*) AS num_flights
   FROM flights GROUP BY origin, dest
   HAVING AVG(distance/(nullif(air_time,0)/60)) > 575;
```

```
SELECT origin, dest,
      AVG(distance/(nullif(air_time,0)/60) AS flight_speed),
      COUNT(*) AS num_flights
   FROM flights GROUP BY origin, dest
   HAVING AVG(flight_speed) > 575;
```



Apache Hive

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q24: Run the following query, then answer the question below. (Note that this query will also be used in the Discussion Prompt, “The Analytic Journey,” so you might want to go directly to that discussion when you're done here.)

```
SELECT origin, dest,
      AVG(distance/(nullif(air_time,0)/60)) AS avg_flight_speed,
      COUNT(*) AS num_flights
   FROM flights
   GROUP BY origin, dest
   HAVING avg_flight_speed > 575;
```

Which of these origin-destination pairs has highest reported flight speed?

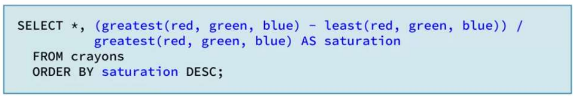
- SLC-SYR
- SLC-BDL
- TUS-RNO
- MCO-JAX

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

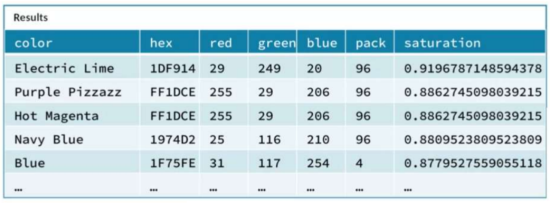
Apache Hive

ORDER BY

- Execution after SELECT FROM WHERE GROUP BY HAVING
- STRING fields → alphabetical order
- Equality of values → arbitrary choice
- Ascending order (ASC) by default, DESC to reverse
- Column(s), expression(s), Column alias, mixture



```
SELECT *, (greatest(red, green, blue) - least(red, green, blue)) /
          greatest(red, green, blue) AS saturation
  FROM crayons
  ORDER BY saturation DESC;
```

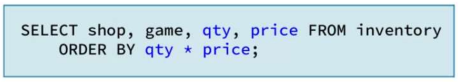


Results

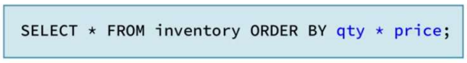
color	hex	red	green	blue	pack	saturation
Electric Lime	1DF914	29	249	20	96	0.9196787148594378
Purple Pizzazz	FF1DCE	255	29	206	96	0.8862745098039215
Hot Magenta	FF1DCE	255	29	206	96	0.8862745098039215
Navy Blue	1974D2	25	116	210	96	0.8809523809523809
Blue	1F75FE	31	117	254	4	0.8779527559055118
…	…	…	…	…	…	…

Hive :
- The column (s) in ORDER BY must be in SELECT
- Columns in an expression (or itself) must be in select

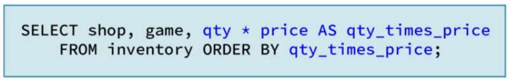
For new versions of Hive:
Shortcuts: SELECT shop, game from inventory order by 2;



```
SELECT shop, game, qty, price FROM inventory
    ORDER BY qty * price;
```



```
SELECT * FROM inventory ORDER BY qty * price;
```

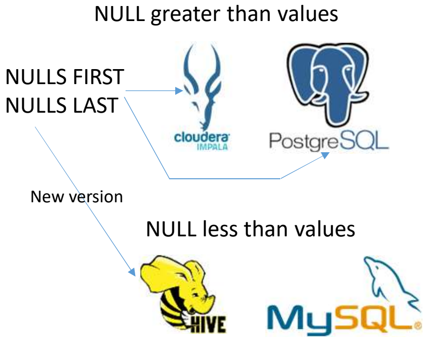


```
SELECT shop, game, qty * price AS qty_times_price
    FROM inventory ORDER BY qty_times_price;
```

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Missing values

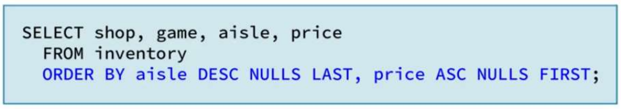


NULL greater than values

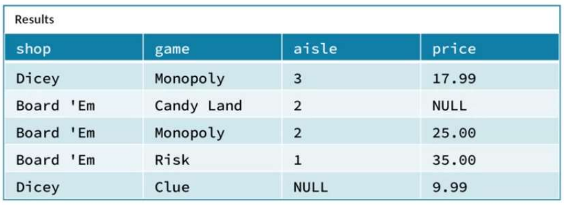
NULLS FIRST
NULLS LAST

New version

NULL less than values



```
SELECT shop, game, aisle, price
  FROM inventory
  ORDER BY aisle DESC NULLS LAST, price ASC NULLS FIRST;
```



Results

shop	game	aisle	price
Dicey	Monopoly	3	17.99
Board 'Em	Candy Land	2	NULL
Board 'Em	Monopoly	2	25.00
Board 'Em	Risk	1	35.00
Dicey	Clue	NULL	9.99

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

LIMIT

- Last Position
- To use with a constant
- Inspect the data
- Avoid returning an exponential number of lines
- Reduce the use of SQL engine resources
- Top n element (beware of the limit)

- Writing order: SELECT FROM WHERE GROUP BY HAVING ORDER BY LIMIT

- Execution order: FROM WHERE GROUP BY HAVING SELECT ORDER BY LIMIT

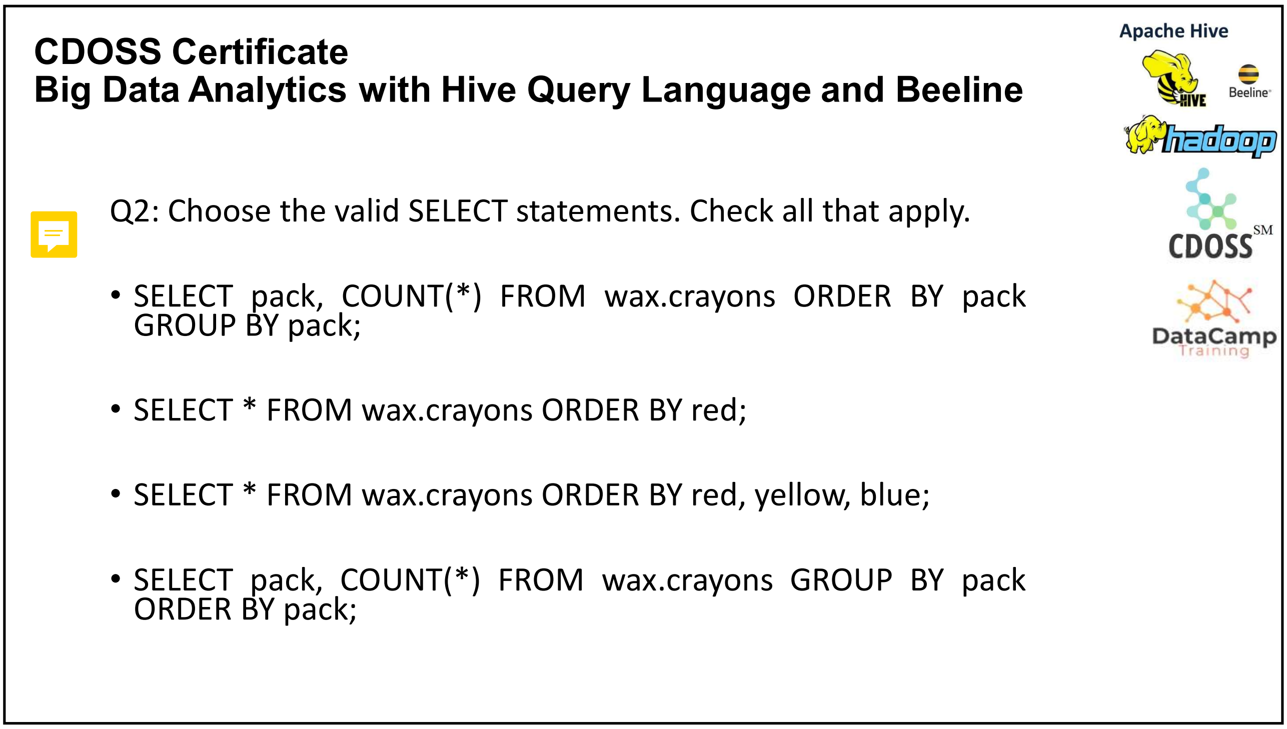
(except when there are aliases detected in SELECT which are in GROUP BY or / ET HAVING)

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q1: Which games might be in the second row of the result set returned by running the query below? Check all that apply. SELECT * FROM games ORDER BY min_age;

- Monopoly
- Risk
- Scrabble
- Candy Land
- Clue

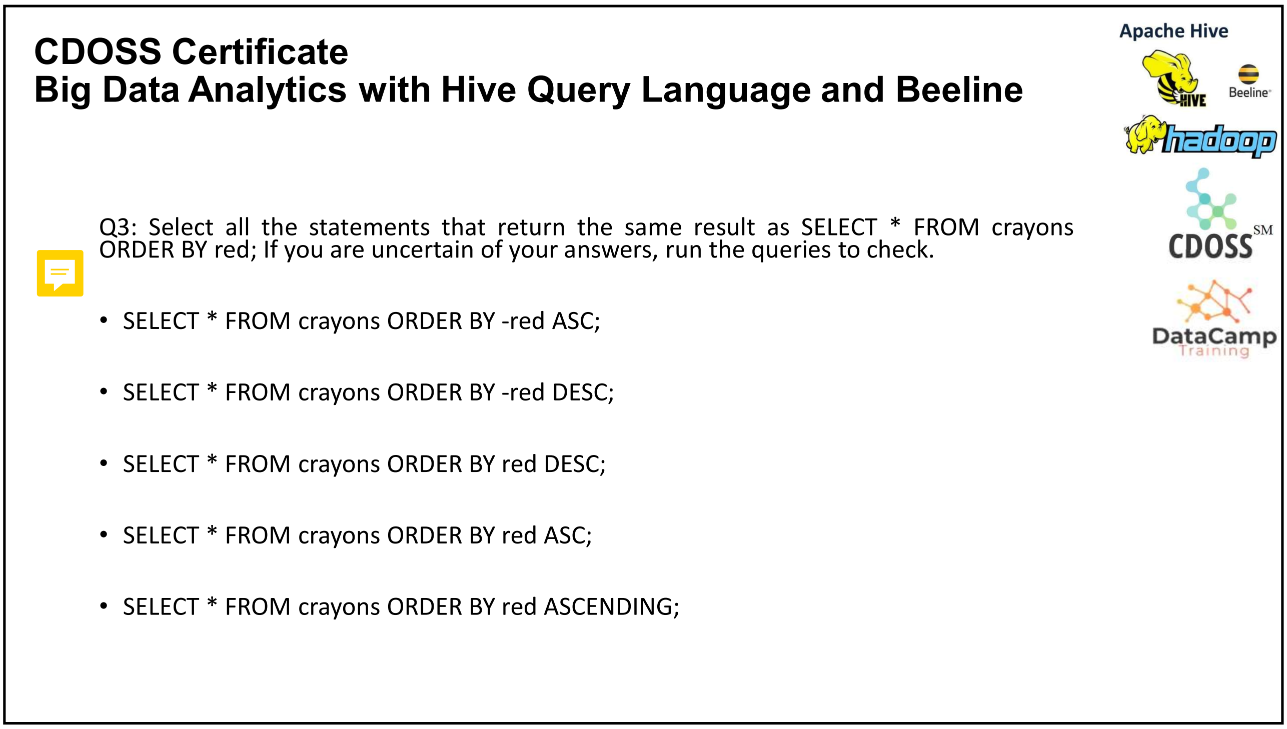


CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q2: Choose the valid SELECT statements. Check all that apply.

- SELECT pack, COUNT(*) FROM wax.crayons ORDER BY pack GROUP BY pack;
- SELECT * FROM wax.crayons ORDER BY red;
- SELECT * FROM wax.crayons ORDER BY red, yellow, blue;
- SELECT pack, COUNT(*) FROM wax.crayons GROUP BY pack ORDER BY pack;



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q3: Select all the statements that return the same result as SELECT * FROM crayons ORDER BY red; If you are uncertain of your answers, run the queries to check.

- SELECT * FROM crayons ORDER BY -red ASC;
- SELECT * FROM crayons ORDER BY -red DESC;
- SELECT * FROM crayons ORDER BY red DESC;
- SELECT * FROM crayons ORDER BY red ASC;
- SELECT * FROM crayons ORDER BY red ASCENDING;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q4: Run the following query with hive and use the result set to answer the question below:

SELECT * FROM wax.crayons ORDER BY pack DESC, red DESC, green ASC;

In the result set, which crayon color is represented in the second row from the top?

- Yellow
- Cotton Candy
- Caribbean Green
- Mountain Meadow
- Canary

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q5: Write and run a SQL query to determine which color in the crayons table has the lowest saturation value, excluding Black and White. The expression to compute saturation is

(greatest(red, green, blue) - least(red, green, blue)) / greatest(red, green, blue)

Which color is it?

- Cadet Blue
- Gray
- Silver
- Timberwolf

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q6: Select the queries that will run without error in Hive. Check all that apply.

- SELECT * FROM wax.crayons ORDER BY pack;
- SELECT color, red + green + blue AS rgb_sum FROM wax.crayons ORDER BY rgb_sum;
- SELECT color, red, green, blue FROM wax.crayons ORDER BY red + green + blue;
- SELECT color FROM wax.crayons ORDER BY pack;
- SELECT color, red + green + blue AS rgb_sum FROM wax.crayons ORDER BY red, green, blue;
- SELECT color, pack FROM wax.crayons ORDER BY pack;

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q7: Select the valid SQL queries. Check all that apply.

```
- SELECT month, AVG(dep_delay) AS avg_dep_delay  FROM flights
   LIMIT 1000 WHERE origin = 'SFO'
   GROUP BY month HAVING avg_dep_delay > 15;
- SELECT month, AVG(dep_delay) AS avg_dep_delay, 10 AS row_limit  FROM flights
   WHERE origin = 'SFO' GROUP BY month
   HAVING avg_dep_delay > 15 LIMIT row_limit;
- SELECT month, AVG(dep_delay) AS avg_dep_delay  FROM flights
   WHERE origin = 'SFO' GROUP BY month
   HAVING avg_dep_delay > 15 LIMIT 1;
- SELECT month, AVG(dep_delay) AS avg_dep_delay  FROM flights
   WHERE origin = 'SFO' LIMIT 100
   GROUP BY month HAVING avg_dep_delay > 15;
- SELECT month, AVG(dep_delay) AS avg_dep_delay  FROM flights
   WHERE origin = 'SFO' GROUP BY month
   HAVING avg_dep_delay > 15 LIMIT -10000;
```

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q8: Select the appropriate uses for the LIMIT clause. Check all that apply:

- Protect against returning an unexpectedly large number of rows
- Randomly sample from a large table
- Return a few rows from a table to inspect some of the values
- Reduce the compute resources used by the SQL engine
- Filter individual rows based on conditions

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q9: The example in this video showed that the routes in the flights table that have the longest average air time are the ones from the New York City airports to Honolulu. The query used in that example was

```
SELECT origin, dest,AVG(air_time) AS avg_air_time,COUNT(air_time) AS count_air_time
   FROM flights  GROUP BY origin, dest
   ORDER BY avg_air_time DESC NULLS LAST  LIMIT 10;
```

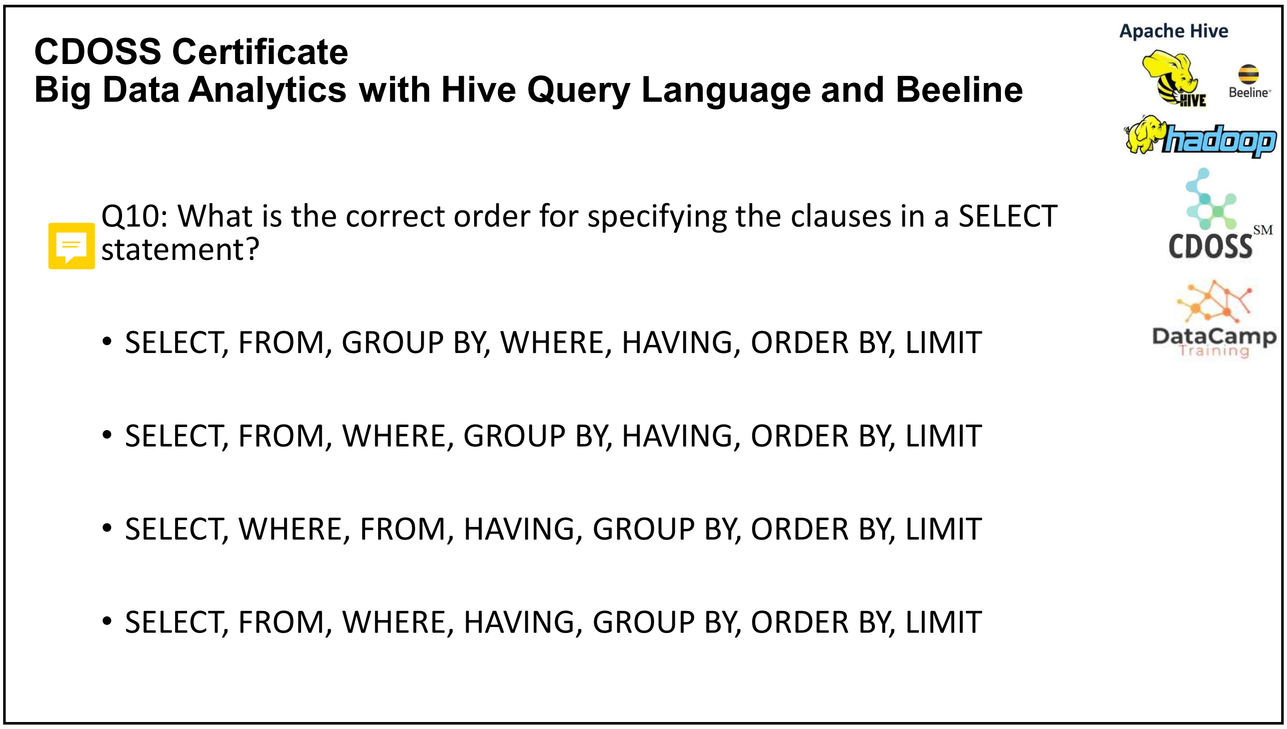
Now, write and run a new query with hive that displays only the two combinations of airline (carrier) and airport (origin) had the quickest flights (smallest average air_time) from New York City to Honolulu. The three New York City airports are EWR, JFK, and LGA. Honolulu airport is HNL.

Select the two correct answers:

- American (AA) flights from LaGuardia (LGA)
- Continental (CO) flights from Newark (EWR)
- Delta (DL) flights from Newark (EWR)
- Delta (DL) flights from Kennedy (JFK)
- Hawaiian (HL) flights from Kennedy (JFK)
- United (UA) flights from Newark (EWR)

The query above should be modified so it ends this way:

```
   WHERE dest = 'HNL' AND origin IN ('EWR', 'JFK', 'LGA')
   GROUP BY origin, carrier
   ORDER BY avg_air_time ASC
   LIMIT 2;
```

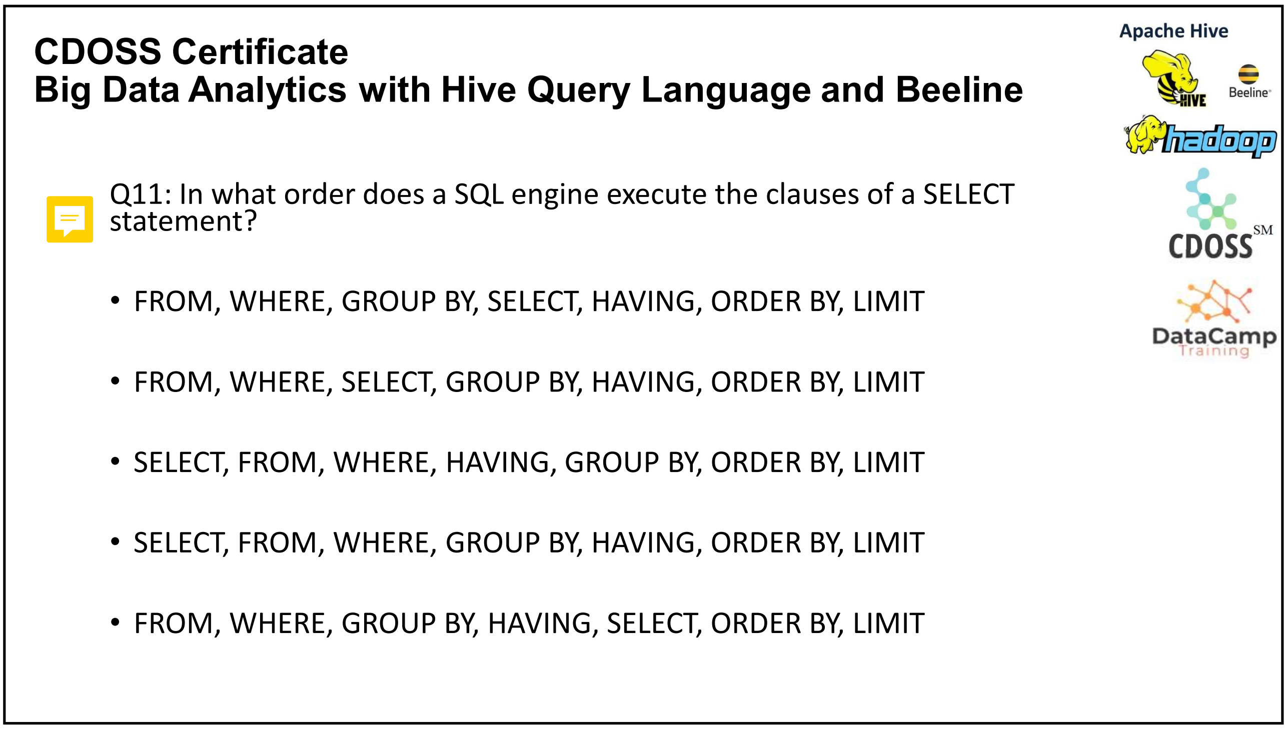


Apache Hive
HIVE
Beeline
hadoop
CDOSS SM
DataCamp
Training

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q10: What is the correct order for specifying the clauses in a SELECT statement?

- SELECT, FROM, GROUP BY, WHERE, HAVING, ORDER BY, LIMIT
- SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY, LIMIT
- SELECT, WHERE, FROM, HAVING, GROUP BY, ORDER BY, LIMIT
- SELECT, FROM, WHERE, HAVING, GROUP BY, ORDER BY, LIMIT



Apache Hive
HIVE
Beeline
hadoop
CDOSS SM
DataCamp
Training

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q11: In what order does a SQL engine execute the clauses of a SELECT statement?

- FROM, WHERE, GROUP BY, SELECT, HAVING, ORDER BY, LIMIT
- FROM, WHERE, SELECT, GROUP BY, HAVING, ORDER BY, LIMIT
- SELECT, FROM, WHERE, HAVING, GROUP BY, ORDER BY, LIMIT
- SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY, LIMIT
- FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY, LIMIT

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

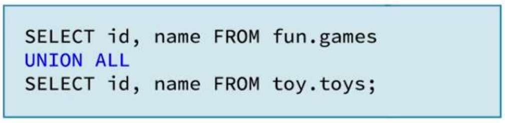
- **UNION**
  - Same field name
  - Same field type
  - Same number of fields

➡ If not, use Aliases and / or cast types

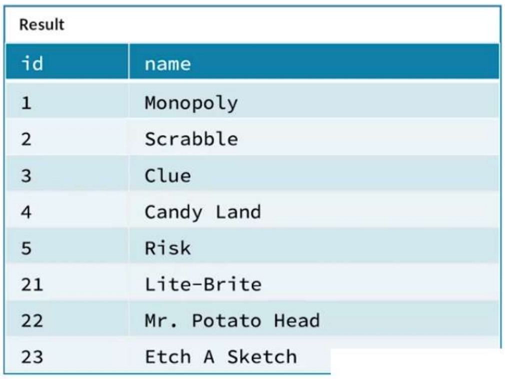
with Hive : UNION (ALL)
Union distinct

with ORDER BY — PostgreSQL ✓ MySQL ✓ cloudera ✗ ⬇ HIVE ✗
Ignored without errors

with LIMIT — PostgreSQL ✓ ⬇ MySQL ✓ cloudera ✗ ⬇ HIVE ✗
it works at the end
We can limit the two selects apart



```
SELECT id, name FROM fun.games
UNION ALL
SELECT id, name FROM toy.toys;
```



Result

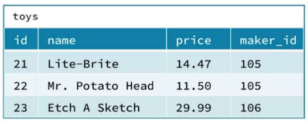
id	name
1	Monopoly
2	Scrabble
3	Clue
4	Candy Land
5	Risk
21	Lite-Brite
22	Mr. Potato Head
23	Etch A Sketch

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

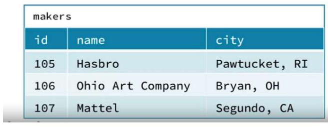
JOIN

```
SELECT *
  FROM toys JOIN makers
    ON toys.maker_id = makers.id;
```



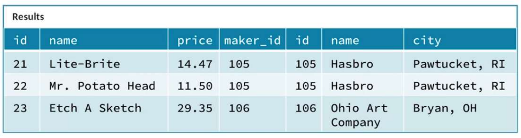
toys

id	name	price	maker_id
21	Lite-Brite	14.47	105
22	Mr. Potato Head	11.50	105
23	Etch A Sketch	29.99	106



makers

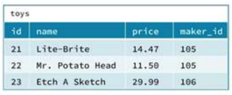
id	name	city
105	Hasbro	Pawtucket, RI
106	Ohio Art Company	Bryan, OH
107	Mattel	Segundo, CA



Results

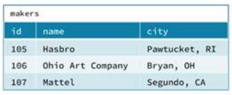
id	name	price	maker_id	id	name	city
21	Lite-Brite	14.47	105	105	Hasbro	Pawtucket, RI
22	Mr. Potato Head	11.50	105	105	Hasbro	Pawtucket, RI
23	Etch A Sketch	29.35	106	106	Ohio Art Company	Bryan, OH

=



toys

id	name	price	maker_id
21	Lite-Brite	14.47	105
22	Mr. Potato Head	11.50	105
23	Etch A Sketch	29.99	106

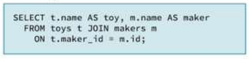


makers

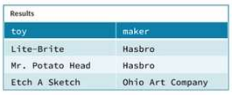
id	name	city
105	Hasbro	Pawtucket, RI
106	Ohio Art Company	Bryan, OH
107	Mattel	Segundo, CA

→ Can be replaced by WHERE :
SELECT …FROM table1,table2 WHERE table1.ch1=table2.ch2

inner join

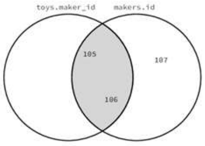


```
SELECT t.name AS toy, m.name AS maker
  FROM toys t JOIN makers m
    ON t.maker_id = m.id;
```



Results

toy	maker
Lite-Brite	Hasbro
Mr. Potato Head	Hasbro
Etch A Sketch	Ohio Art Company



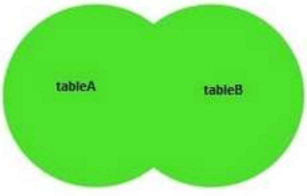
It is possible to join several tables ➡

```
SELECT c.name AS customer_name,
    o.total AS order_total,
    e.first_name AS employee_name
  FROM customers c
    JOIN orders o ON c.cust_id = o.cust_id
    JOIN employees e ON o.empl_id = e.empl_id;
```

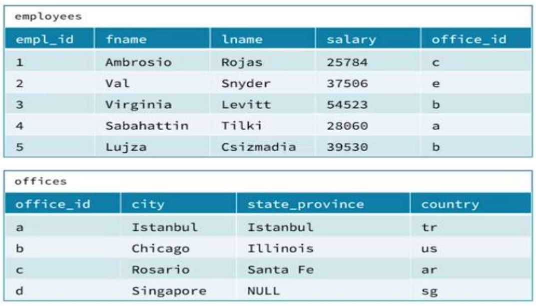
NULLs is not matched: null=null→null
→ We can use 'IS NOT DISTINCT FROM' with WHERE

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive



Full Outer Join

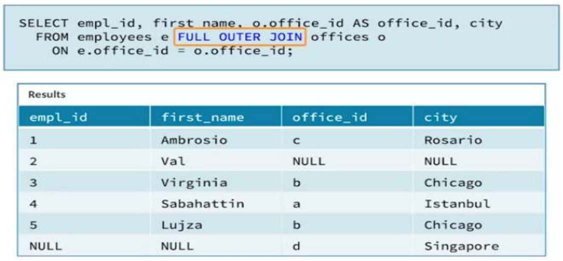


employees

empl_id	fname	lname	salary	office_id
1	Ambrosio	Rojas	25784	c
2	Val	Snyder	37506	e
3	Virginia	Levitt	54523	b
4	Sabahattin	Tilki	28060	a
5	Lujza	Csizmadia	39530	b

offices

office_id	city	state_province	country
a	Istanbul	Istanbul	tr
b	Chicago	Illinois	us
c	Rosario	Santa Fe	ar
d	Singapore	NULL	sg



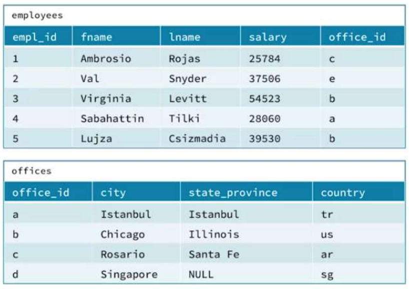
```
SELECT empl_id, first_name, o.office_id AS office_id, city
  FROM employees e FULL OUTER JOIN offices o
    ON e.office_id = o.office_id;
```

Results

empl_id	first_name	office_id	city
1	Ambrosio	c	Rosario
2	Val	NULL	NULL
3	Virginia	b	Chicago
4	Sabahattin	a	Istanbul
5	Lujza	b	Chicago
NULL	NULL	d	Singapore

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

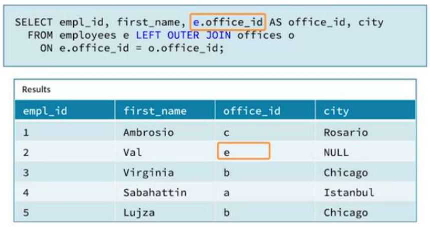


employees

empl_id	fname	lname	salary	office_id
1	Ambrosio	Rojas	25784	c
2	Val	Snyder	37506	e
3	Virginia	Levitt	54523	b
4	Sabahattin	Tilki	28060	a
5	Lujza	Csizmadia	39530	b

offices

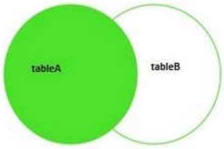
office_id	city	state_province	country
a	Istanbul	Istanbul	tr
b	Chicago	Illinois	us
c	Rosario	Santa Fe	ar
d	Singapore	NULL	sg



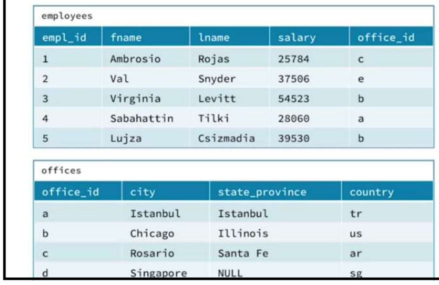
```
SELECT empl_id, first_name, e.office_id AS office_id, city
  FROM employees e LEFT OUTER JOIN offices o
    ON e.office_id = o.office_id;
```

Results

empl_id	first_name	office_id	city
1	Ambrosio	c	Rosario
2	Val	e	NULL
3	Virginia	b	Chicago
4	Sabahattin	a	Istanbul
5	Lujza	b	Chicago



Left Outer Join

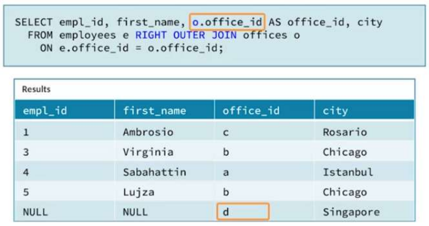


employees

empl_id	fname	lname	salary	office_id
1	Ambrosio	Rojas	25784	c
2	Val	Snyder	37506	e
3	Virginia	Levitt	54523	b
4	Sabahattin	Tilki	28060	a
5	Lujza	Csizmadia	39530	b

offices

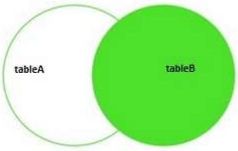
office_id	city	state_province	country
a	Istanbul	Istanbul	tr
b	Chicago	Illinois	us
c	Rosario	Santa Fe	ar
d	Singapore	NULL	sg



```
SELECT empl_id, first_name, o.office_id AS office_id, city
  FROM employees e RIGHT OUTER JOIN offices o
    ON e.office_id = o.office_id;
```

Results

empl_id	first_name	office_id	city
1	Ambrosio	c	Rosario
3	Virginia	b	Chicago
4	Sabahattin	a	Istanbul
5	Lujza	b	Chicago
NULL	NULL	d	Singapore



Right Outer Join

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Specific joins

Cross join :
Cartesian product (JOIN without on and without WHERE)

No equijoin : < > <>

PostgreSQL	MySQL	cloudera	HIVE
✓	✓	✓	✗

```
SELECT first_name, last_name, grade
FROM employees e JOIN salary_grades g
ON e.salary >= g.min_salary AND e.salary <= g.max_salary;
```

employees

empl_id	fname	lname	salary	office_id
1	Ambrosio	Rojas	25784	c
2	Val	Snyder	37506	e
3	Virginia	Levitt	54523	b
4	Sabahattin	Tilki	28060	a
5	Lujza	Csizmadia	39530	b

salary_grades

grade	min_salary	max_salary
1	10000	19999
2	20000	29999
3	30000	39999
4	40000	49999
5	50000	59999

Results

first_name	last_name	grade
Ambrosio	Rojas	2
Val	Snyder	3
Virginia	Levitt	5
Sabahattin	Tilki	2
Lujza	Csizmadia	3

Left semi-join : join with filter

```
SELECT DISTINCT manufacturer, model
  FROM planes p LEFT SEMI JOIN flights f
    ON p.tailnum = f.tailnum AND f.distance > 4000 * 1.15;
```

Results

manufacturer	model
BOEING	777-300ER
AIRBUS	A330-243
BOEING	767-424ER
BOEING	767-324
BOMBARDIER INC	CL-600-2B19
BOEING	777-222
…	…

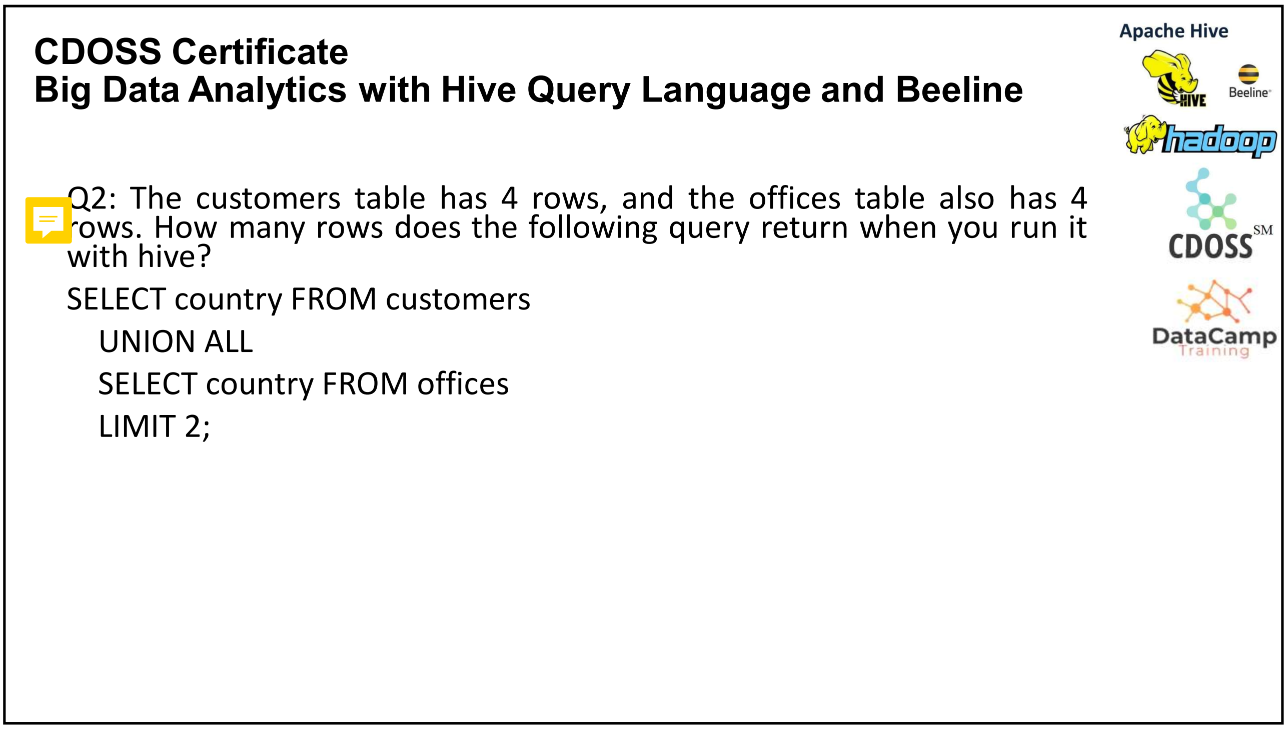
CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q1: Which value is guaranteed to be in the top row of the result set when you run the following query with hive?

```
SELECT country FROM customers
   UNION ALL
   SELECT country FROM offices
   ORDER BY country DESC;
```

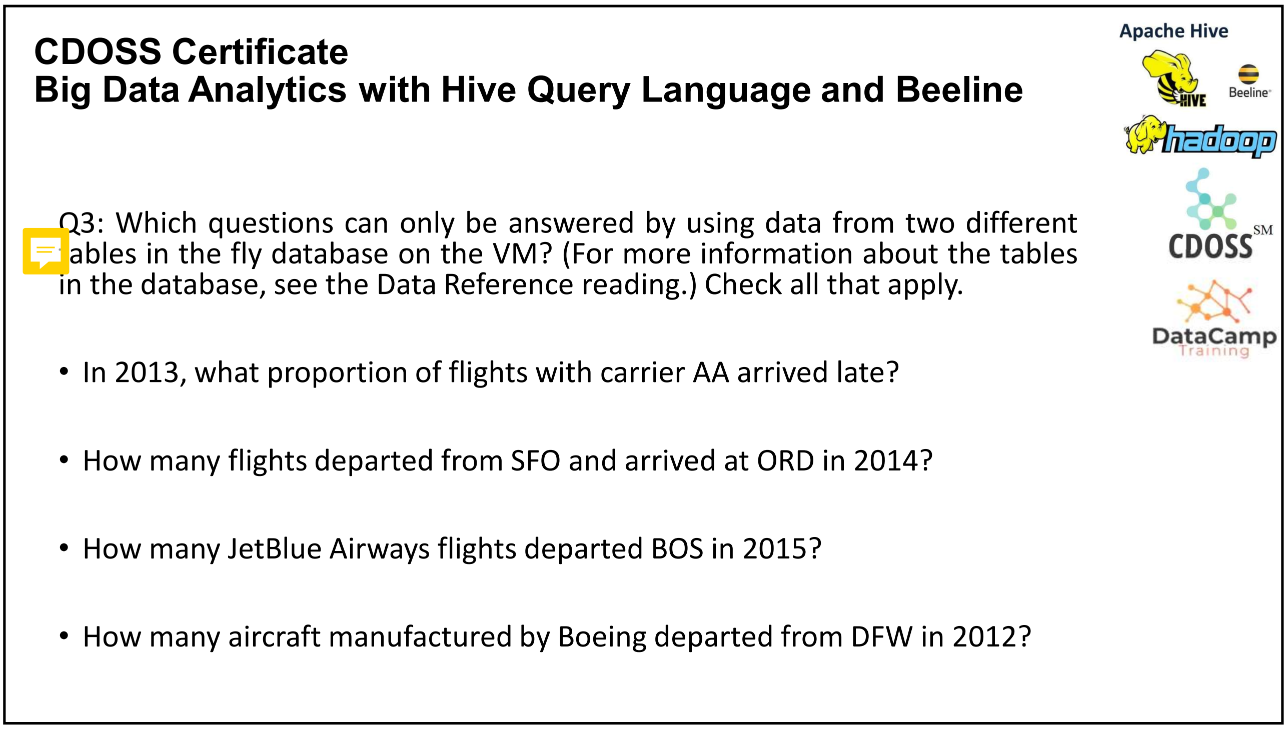
- ar
- ja
- pk
- ug
- us
- No particular value is guaranteed to be in the top row



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q2: The customers table has 4 rows, and the offices table also has 4 rows. How many rows does the following query return when you run it with hive?

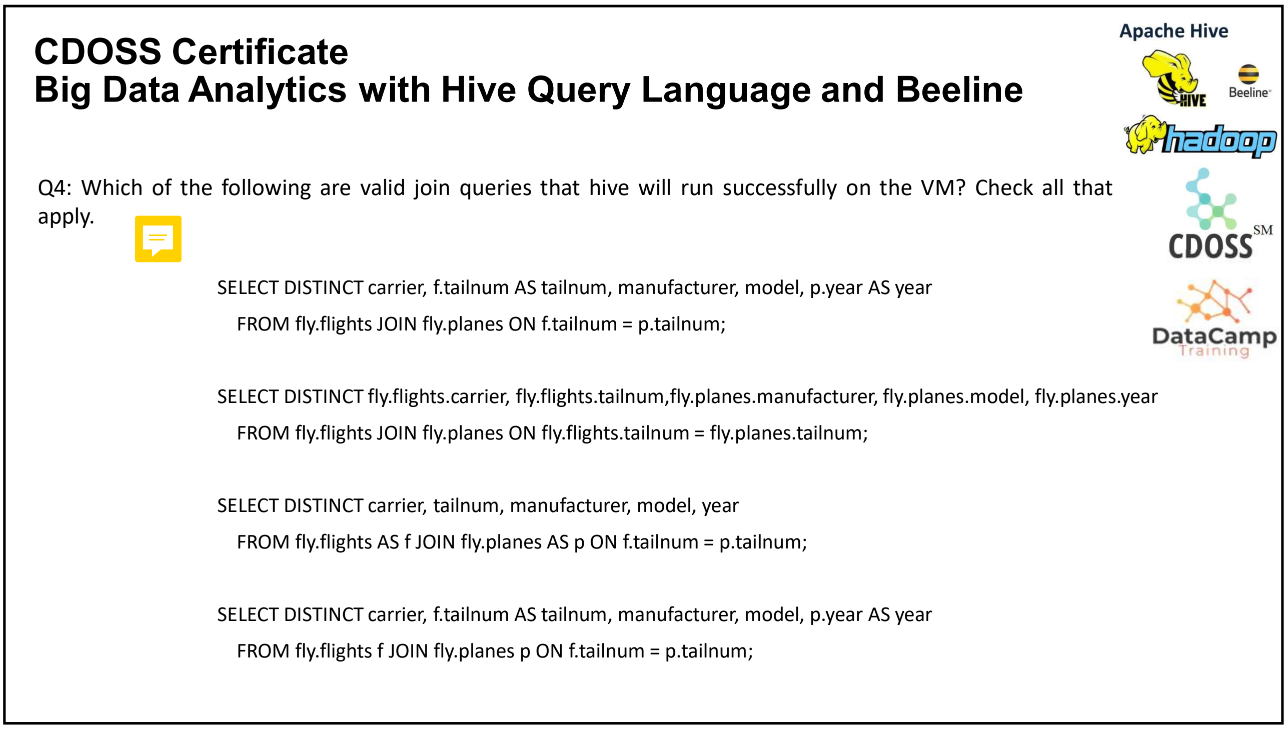
```
SELECT country FROM customers
   UNION ALL
   SELECT country FROM offices
   LIMIT 2;
```



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q3: Which questions can only be answered by using data from two different tables in the fly database on the VM? (For more information about the tables in the database, see the Data Reference reading.) Check all that apply.

- In 2013, what proportion of flights with carrier AA arrived late?
- How many flights departed from SFO and arrived at ORD in 2014?
- How many JetBlue Airways flights departed BOS in 2015?
- How many aircraft manufactured by Boeing departed from DFW in 2012?



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

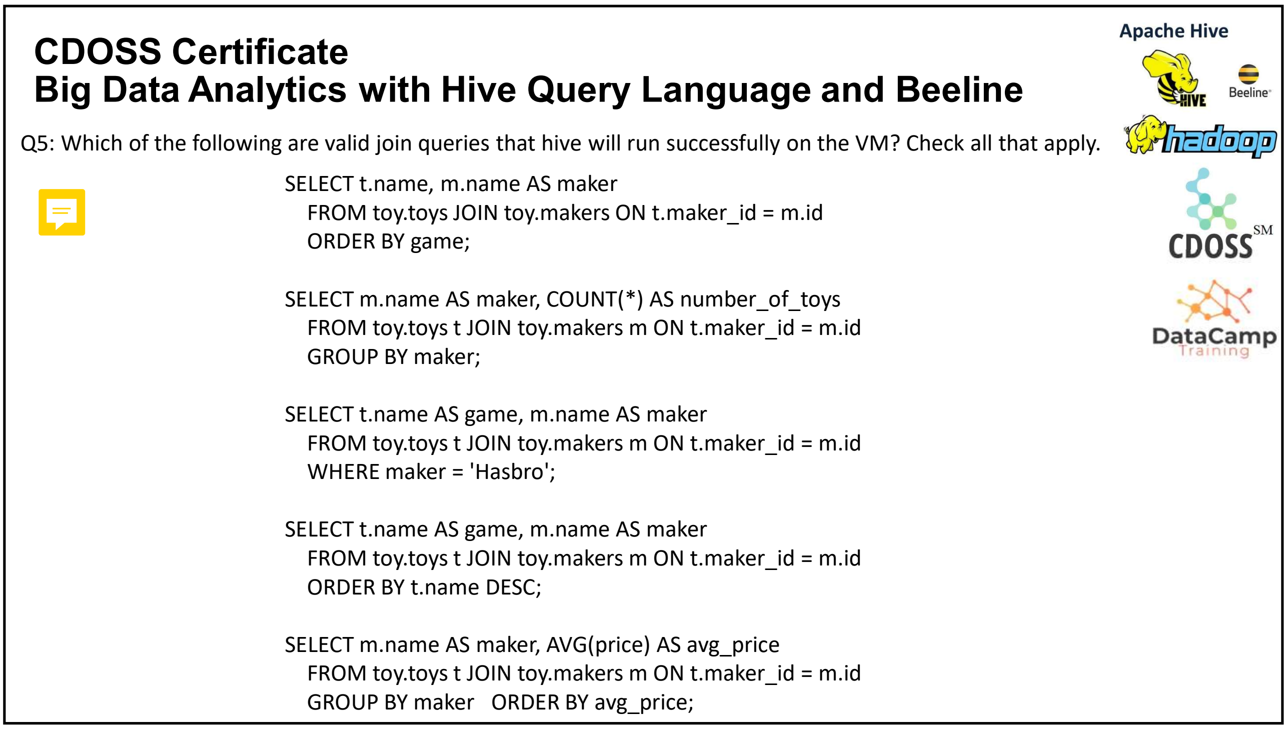
Q4: Which of the following are valid join queries that hive will run successfully on the VM? Check all that apply.

```
SELECT DISTINCT carrier, f.tailnum AS tailnum, manufacturer, model, p.year AS year
   FROM fly.flights JOIN fly.planes ON f.tailnum = p.tailnum;

SELECT DISTINCT fly.flights.carrier, fly.flights.tailnum,fly.planes.manufacturer, fly.planes.model, fly.planes.year
   FROM fly.flights JOIN fly.planes ON fly.flights.tailnum = fly.planes.tailnum;

SELECT DISTINCT carrier, tailnum, manufacturer, model, year
   FROM fly.flights AS f JOIN fly.planes AS p ON f.tailnum = p.tailnum;

SELECT DISTINCT carrier, f.tailnum AS tailnum, manufacturer, model, p.year AS year
   FROM fly.flights f JOIN fly.planes p ON f.tailnum = p.tailnum;
```



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q5: Which of the following are valid join queries that hive will run successfully on the VM? Check all that apply.

```
SELECT t.name, m.name AS maker
   FROM toy.toys JOIN toy.makers ON t.maker_id = m.id
   ORDER BY game;

SELECT m.name AS maker, COUNT(*) AS number_of_toys
   FROM toy.toys t JOIN toy.makers m ON t.maker_id = m.id
   GROUP BY maker;

SELECT t.name AS game, m.name AS maker
   FROM toy.toys t JOIN toy.makers m ON t.maker_id = m.id
   WHERE maker = 'Hasbro';

SELECT t.name AS game, m.name AS maker
   FROM toy.toys t JOIN toy.makers m ON t.maker_id = m.id
   ORDER BY t.name DESC;

SELECT m.name AS maker, AVG(price) AS avg_price
   FROM toy.toys t JOIN toy.makers m ON t.maker_id = m.id
   GROUP BY maker   ORDER BY avg_price;
```

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q6: Review the contents of the employees and offices tables (see the Data Reference reading), and try to answer the following question without actually running the join query. (You can check your answers by running the query in hive and viewing the result set.)

```
SELECT first_name, last_name, city
   FROM employees e INNER JOIN offices o
      ON e.office_id = o.office_id;
```

Which employees are included in the inner join result? Check all that apply.

- Ambrosio Rojas
- Val Snyder
- Virginia Levitt
- Sabahattin Tilki
- Lujza Csizmadia

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Q7: Review the contents of the employees and offices tables and try to answer the following question without actually running the join query. (You can check your answers by running the query in hive and viewing the result set.)

```
SELECT first_name, last_name, city
   FROM employees e INNER JOIN offices o
      ON e.office_id = o.office_id;
```

Which offices are included in the inner join result? Check all that apply.

- Istanbul
- Chicago
- Rosario
- Singapore

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q8: Which FROM clauses could you use to return data about all the customers, even the ones who have not placed any orders? Select all that apply.

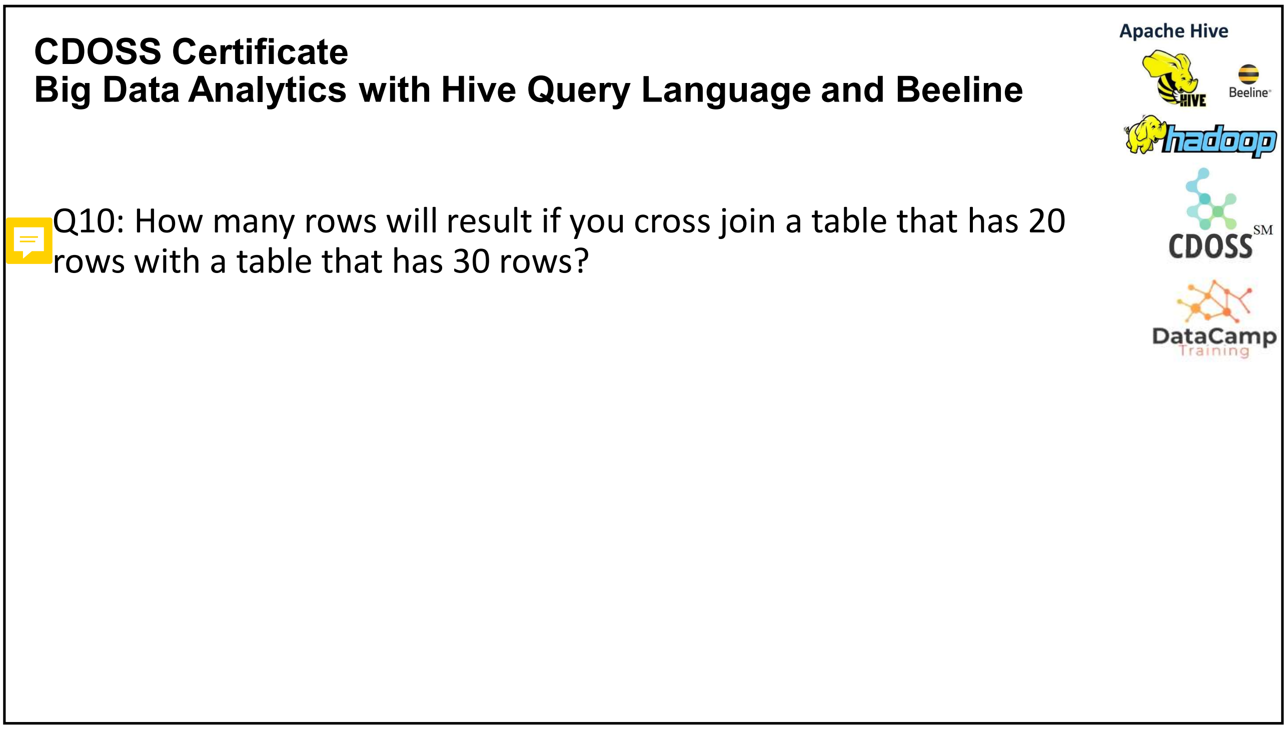
- FROM orders o LEFT OUTER JOIN customers c ON o.cust_id = c.cust_id
- FROM customers c RIGHT OUTER JOIN orders o ON c.cust_id = o.cust_id
- FROM customers c LEFT OUTER JOIN orders o ON c.cust_id = o.cust_id
- FROM orders o RIGHT OUTER JOIN customers c ON o.cust_id = c.cust_id

CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q9: Which of the following queries returns only the employees whose office IDs do not match any office IDs found in the offices table?

```
SELECT empl_id, first_name, last_name
  FROM employees e LEFT OUTER JOIN offices o ON e.office_id = o.office_id
  WHERE office_id IS NULL;
SELECT empl_id, first_name, last_name
  FROM offices o LEFT OUTER JOIN employees e ON e.office_id = o.office_id
  WHERE e.office_id IS NULL;
SELECT empl_id, first_name, last_name
  FROM employees e LEFT OUTER JOIN offices o ON e.office_id = o.office_id
  WHERE o.office_id IS NULL;
SELECT empl_id, first_name, last_name
  FROM employees e LEFT OUTER JOIN offices o ON e.office_id = o.office_id
  WHERE e.office_id IS NULL;
SELECT empl_id, first_name, last_name
  FROM offices o LEFT OUTER JOIN employees e ON e.office_id = o.office_id
  WHERE o.office_id IS NULL;
```



CDOSS Certificate
Big Data Analytics with Hive Query Language and Beeline

Apache Hive

Q10: How many rows will result if you cross join a table that has 20 rows with a table that has 30 rows?