**## Spark and SQL Interview Scenerio Questions**

**### Table of Contents**

|No| Scenerios                                                                |

|--|--------------------------------------------------------------------------|

|1 |[Scenerio-1](#scenerio-1)                                                 |

|2 |[Scenerio-2](#scenerio-2)                                                 |

|3 |[Scenerio-3](#scenerio-3)                                                 |

|4 |[Scenerio-4](#scenerio-4)                                                 |

|5 |[Scenerio-5](#scenerio-5)                                                 |

|6 |[Scenerio-6](#scenerio-6)                                                 |

|7 |[Scenerio-7](#scenerio-7)                                                 |

|8 |[Scenerio-8](#scenerio-8)                                                 |

|9 |[Scenerio-9](#scenerio-9)                                                 |

|10|[Scenerio-10](#scenerio-10)                                               |

|11|[Scenerio-11](#scenerio-11)                                               |

|12|[Scenerio-12](#scenerio-12)                                               |

|13|[Scenerio-13](#scenerio-13)                                               |

|14|[Scenerio-14](#scenerio-14)                                               |

|15|[Scenerio-15](#scenerio-15)                                               |

|16|[Scenerio-16](#scenerio-16)                                               |

|17|[Scenerio-17](#scenerio-17)                                               |

|18|[Scenerio-18](#scenerio-18)                                               |

|19|[Scenerio-19](#scenerio-19)                                               |

|20|[Scenerio-20](#scenerio-20)                                               |

|21|[Scenerio-21](#scenerio-21)                                               |

|22|[Scenerio-22](#scenerio-22)                                               |

|23|[Scenerio-23](#scenerio-23)                                               |

|24|[Scenerio-24](#scenerio-24)                                               |

|25|[Scenerio-25](#scenerio-25)                                               |

|26|[Scenerio-26](#scenerio-26)                                               |

|27|[Scenerio-27](#scenerio-27)                                               |

|28|[Scenerio-28](#scenerio-28)                                               |

|29|[Scenerio-29](#scenerio-29)                                               |

|30|[Scenerio-30](#scenerio-30)                                               |

|31|[Scenerio-31](#scenerio-31)                                               |

|32|[Scenerio-32](#scenerio-32)                                               |

|33|[Scenerio-33](#scenerio-33)                                               |

|34|[Scenerio-34](#scenerio-34)                                               |

|35|[Scenerio-35](#scenerio-35)                                               |

|36|[Scenerio-36](#scenerio-36)                                               |

**### Scenerio-1**

**#### Query to get who are getting equal salary**

**#### Input :-**

```

+--------+---------+--------+------+-------------------+------+

|workerid|firstname|lastname|salary|        joiningdate|depart|

+--------+---------+--------+------+-------------------+------+

|     001|   Monika|   Arora|100000|2014-02-20 09:00:00|    HR|

|     002| Niharika|   Verma|300000|2014-06-11 09:00:00| Admin|

|     003|   Vishal| Singhal|300000|2014-02-20 09:00:00|    HR|

|     004|  Amitabh|   Singh|500000|2014-02-20 09:00:00| Admin|

|     005|    Vivek|   Bhati|500000|2014-06-11 09:00:00| Admin|

+--------+---------+--------+------+-------------------+------+

```

**#### Expected Output :-**

```

+--------+---------+--------+------+-------------------+------+

|workerid|firstname|lastname|salary|        joiningdate|depart|

+--------+---------+--------+------+-------------------+------+

|     002| Niharika|   Verma|300000|2014-06-11 09:00:00| Admin|

|     003|   Vishal| Singhal|300000|2014-02-20 09:00:00|    HR|

|     004|  Amitabh|   Singh|500000|2014-02-20 09:00:00| Admin|

|     005|    Vivek|   Bhati|500000|2014-06-11 09:00:00| Admin|

+--------+---------+--------+------+-------------------+------+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenarios-spark-sql/blob/master/src/pack/Scenerio1.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenarios-spark-sql/blob/master/Scenerio-1.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-2**

**#### (Need the dates when the status gets changed like ordered to dispatched)**

**#### Input :-**

```

+-------+----------+----------+

|orderid|statusdate|    status|

+-------+----------+----------+

|      1|     1-Jan|   Ordered|

|      1|     2-Jan|dispatched|

|      1|     3-Jan|dispatched|

|      1|     4-Jan|   Shipped|

|      1|     5-Jan|   Shipped|

|      1|     6-Jan| Delivered|

|      2|     1-Jan|   Ordered|

|      2|     2-Jan|dispatched|

|      2|     3-Jan|   shipped|

+-------+----------+----------+

```

**#### Expected Output :-**

```

+-------+----------+----------+

|orderid|statusdate|    status|

+-------+----------+----------+

|      1|     2-Jan|dispatched|

|      1|     3-Jan|dispatched|

|      2|     2-Jan|dispatched|

+-------+----------+----------+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenarios-spark-sql/blob/master/src/pack/Scenerio2.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenarios-spark-sql/blob/master/Scenerio2.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-3**

**#### Input :-**

```

+--------+----------+------+

|sensorid| timestamp|values|

+--------+----------+------+

|    1111|2021-01-15|    10|

|    1111|2021-01-16|    15|

|    1111|2021-01-17|    30|

|    1112|2021-01-15|    10|

|    1112|2021-01-15|    20|

|    1112|2021-01-15|    30|

+--------+----------+------+

```

**#### Expected Output :-**

```

+--------+----------+------+

|sensorid| timestamp|values|

+--------+----------+------+

|    1111|2021-01-15|     5|

|    1111|2021-01-16|    15|

|    1112|2021-01-15|    10|

|    1112|2021-01-15|    10|

+--------+----------+------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio3.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio3.py>) <br>

SQL -

```

SELECT sensorid,

       timestamp,

       ( newvalues - values ) AS values

FROM  (SELECT \*,

              Lead(values, 1, 0)

                OVER(

                  partition BY sensorid

                  ORDER BY values) AS newvalues

       FROM   timetab)

WHERE  newvalues != 0

```

Pandas -

```

import pandas as pd

data = [

    (1111, "2021-01-15", 10),

    (1111, "2021-01-16", 15),

    (1111, "2021-01-17", 30),

    (1112, "2021-01-15", 10),

    (1112, "2021-01-15", 20),

    (1112, "2021-01-15", 30),

]

df = pd.DataFrame(data, columns=["sensorid", "timestamp", "values"])

print(df)

df["newvalues"] = df.groupby("sensorid")["values"].shift(-1)

print(df)

df = df.dropna(subset=["newvalues"])

print(df)

df["values"] = df["newvalues"] - df["values"]

print(df)

df = df.drop(columns=["newvalues"])

print(df)

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-4**

**#### (Write a query to list the unique customer names in the custtab table, along with the number of addresses associated with each customer.)**

**#### Input :-**

```

+------+-----------+-------+

|custid|   custname|address|

+------+-----------+-------+

|     1|   Mark Ray|     AB|

|     2|Peter Smith|     CD|

|     1|   Mark Ray|     EF|

|     2|Peter Smith|     GH|

|     2|Peter Smith|     CD|

|     3|       Kate|     IJ|

+------+-----------+-------+

```

**#### Expected Output :-**

```

+------+-----------+--------+

|custid|   custname| address|

+------+-----------+--------+

|     1|   Mark Ray|[EF, AB]|

|     2|Peter Smith|[CD, GH]|

|     3|       Kate|    [IJ]|

+------+-----------+--------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio4.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio4.py>) <br>

SQL -

```

SELECT custid,

       custname,

       Collect\_set(address) AS address

FROM   custtab

GROUP  BY custid,

          custname

ORDER  BY custid

```

Pandas -

```

data = [

    (1, "Mark Ray", "AB"),

    (2, "Peter Smith", "CD"),

    (1, "Mark Ray", "EF"),

    (2, "Peter Smith", "GH"),

    (2, "Peter Smith", "CD"),

    (3, "Kate", "IJ"),

]

df = pd.DataFrame(data, columns=["custid", "custname", "address"])

print(df)

finaldf = (

    df.groupby(["custid", "custname"])["address"]

    .apply(lambda x: list(set(x)))

    .reset\_index()

)

print(finaldf)

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-5**

\* Read data from above file into dataframes(df1 and df2).

\* Display number of partitions in df1.

\* Create a new dataframe df3 from df1, along with a new column salary, and keep it constant 1000

\* append df2 and df3, and form df4

\* Remove records which have invalid email from df4, emails with @ are considered to be valid.

\* Write df4 to a target location, by partitioning on salary.

**#### Input :-**

```

+---+----+---+-------------+

| id|name|age|        email|

+---+----+---+-------------+

|  1| abc| 31|abc@gmail.com|

|  2| def| 23| defyahoo.com|

|  3| xyz| 26|xyz@gmail.com|

|  4| qwe| 34| qwegmail.com|

|  5| iop| 24|iop@gmail.com|

+---+----+---+-------------+

```

```

+---+----+---+---------------+------+

| id|name|age|          email|salary|

+---+----+---+---------------+------+

| 11| jkl| 22|  abc@gmail.com|  1000|

| 12| vbn| 33|  vbn@yahoo.com|  3000|

| 13| wer| 27|            wer|  2000|

| 14| zxc| 30|        zxc.com|  2000|

| 15| lkj| 29|lkj@outlook.com|  2000|

+---+----+---+---------------+------+

```

**#### Expected Output :-**

```

+---+----+---+---------------+------+

| id|name|age|          email|salary|

+---+----+---+---------------+------+

|  1| abc| 31|  abc@gmail.com|  1000|

|  3| xyz| 26|  xyz@gmail.com|  1000|

|  5| iop| 24|  iop@gmail.com|  1000|

| 11| jkl| 22|  abc@gmail.com|  1000|

| 12| vbn| 33|  vbn@yahoo.com|  3000|

| 15| lkj| 29|lkj@outlook.com|  2000|

+---+----+---+---------------+------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio5.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio5.py>) <br>

Pandas -

```

import pandas as pd

# Read data convert into dataframes(df1 and df2).

data1 = [

    (1, "abc", 31, "abc@gmail.com"),

    (2, "def", 23, "defyahoo.com"),

    (3, "xyz", 26, "xyz@gmail.com"),

    (4, "qwe", 34, "qwegmail.com"),

    (5, "iop", 24, "iop@gmail.com"),

]

df1 = pd.DataFrame(data1, columns=["id", "name", "age", "email"])

print(df1)

data2 = [

    (11, "jkl", 22, "abc@gmail.com", 1000),

    (12, "vbn", 33, "vbn@yahoo.com", 3000),

    (13, "wer", 27, "wer", 2000),

    (14, "zxc", 30, "zxc.com", 2000),

    (15, "lkj", 29, "lkj@outlook.com", 2000),

]

df2 = pd.DataFrame(data2, columns=["id", "name", "age", "email", "salary"])

print(df2)

# Create a new dataframe df3 from df1, along with a new column salary, and keep it constant 1000

df3 = df1.copy()

df3["salary"] = 1000

print(df3)

# append df2 and df3, and form df4

df4 = pd.concat([df2, df3])

df4 = df4.sort\_values("id")

print(df4)

# Remove records which have invalid email from df4, emails with @ are considered to be valid.

finaldf = df4[df4["email"].str.contains("@", na=False)]

print(finaldf)

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-6**

**#### (For Employee salary greater than 10000 give designation as manager else employee)**

**#### Input :-**

```

+-----+----+------+

|empid|name|salary|

+-----+----+------+

|    1|   a| 10000|

|    2|   b|  5000|

|    3|   c| 15000|

|    4|   d| 25000|

|    5|   e| 50000|

|    6|   f|  7000|

+-----+----+------+

```

**#### Expected Output :-**

```

+-----+----+------+-----------+

|empid|name|salary|Designation|

+-----+----+------+-----------+

|    1|   a| 10000|   Employee|

|    2|   b|  5000|   Employee|

|    3|   c| 15000|    Manager|

|    4|   d| 25000|    Manager|

|    5|   e| 50000|    Manager|

|    6|   f|  7000|   Employee|

+-----+----+------+-----------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio6.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio6.py>) <br>

SQL -

```

SELECT \*,

    CASE

    WHEN salary > 10000 THEN

    'Manager'

    ELSE 'Employee'

    END AS Designation

FROM emptab

```

Pandas -

```

import pandas as pd

data = [

    ("1", "a", 10000),

    ("2", "b", 5000),

    ("3", "c", 15000),

    ("4", "d", 25000),

    ("5", "e", 50000),

    ("6", "f", 7000),

]

df = pd.DataFrame(data, columns=["empid", "name", "salary"])

print(df)

def emp\_desgnination(salary):

    return "Manager" if salary > 10000 else "Employee"

df["Desgniation"] = df["salary"].apply(emp\_desgnination)

print(df)

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-7**

**#### Input :-**

```

+-------+----------+----+--------+-----+

|sale\_id|product\_id|year|quantity|price|

+-------+----------+----+--------+-----+

|      1|       100|2010|      25| 5000|

|      2|       100|2011|      16| 5000|

|      3|       100|2012|       8| 5000|

|      4|       200|2010|      10| 9000|

|      5|       200|2011|      15| 9000|

|      6|       200|2012|      20| 7000|

|      7|       300|2010|      20| 7000|

|      8|       300|2011|      18| 7000|

|      9|       300|2012|      20| 7000|

+-------+----------+----+--------+-----+

```

**#### Expected Output :-**

```

+-------+----------+----+--------+-----+

|sale\_id|product\_id|year|quantity|price|

+-------+----------+----+--------+-----+

|      6|       200|2012|      20| 7000|

|      9|       300|2012|      20| 7000|

|      1|       100|2010|      25| 5000|

|      8|       300|2011|      18| 7000|

+-------+----------+----+--------+-----+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio7.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio7.py>) <br>

SQL -

```

SELECT

  \*

FROM

  (

    SELECT

      \*,

      DENSE\_RANK() OVER (

        PARTITION BY year

        ORDER BY

          quantity DESC

      ) AS rank

    FROM

      salestab

  ) AS rankdf

WHERE

  rank = 1

ORDER BY

  sale\_id

```

Pandas -

```

import pandas as pd

data = [

    (1, 100, 2010, 25, 5000),

    (2, 100, 2011, 16, 5000),

    (3, 100, 2012, 8, 5000),

    (4, 200, 2010, 10, 9000),

    (5, 200, 2011, 15, 9000),

    (6, 200, 2012, 20, 7000),

    (7, 300, 2010, 20, 7000),

    (8, 300, 2011, 18, 7000),

    (9, 300, 2012, 20, 7000),

]

df = pd.DataFrame(data, columns=["sale\_id", "product\_id", "year", "quantity", "price"])

print(df)

df["rank"] = df.groupby("year")["quantity"].rank(method="dense", ascending=False)

print(df)

df = df[df["rank"] == 1]

print(df)

df = df.drop("rank", axis=1).sort\_values("sale\_id")

print(df)

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-8**

**#### Input :-**

```

+--------+

|   teams|

+--------+

|   India|

|Pakistan|

|SriLanka|

+--------+

```

**#### Expected Output :-**

```

+--------------------+

|             matches|

+--------------------+

|   India Vs Pakistan|

|   India Vs SriLanka|

|Pakistan Vs SriLanka|

+--------------------+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio8.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio8.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-9**

**#### (write spark code, list of name of participants who has rank=1 most number of times)**

**#### Input :-**

```

+----+---------------+

|name|           rank|

+----+---------------+

|   a|   [1, 1, 1, 3]|

|   b|   [1, 2, 3, 4]|

|   c|[1, 1, 1, 1, 4]|

|   d|            [3]|

+----+---------------+

```

**#### Expected Output :-**

```

c

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio9.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio9.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-10**

**#### Input :-**

```

+-----+-------------+-------------+

|empid|commissionamt|monthlastdate|

+-----+-------------+-------------+

|    1|          300|  31-Jan-2021|

|    1|          400|  28-Feb-2021|

|    1|          200|  31-Mar-2021|

|    2|         1000|  31-Oct-2021|

|    2|          900|  31-Dec-2021|

+-----+-------------+-------------+

```

**#### Expected Output :-**

```

+-----+-------------+-------------+

|empid|commissionamt|monthlastdate|

+-----+-------------+-------------+

|    1|          200|  31-Mar-2021|

|    2|         1000|  31-Oct-2021|

+-----+-------------+-------------+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio10.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio10.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-11**

**#### (I have a table called Emp\_table, it has 3 columns, Emp name, emp ID , salary**

in this I want to get salaries that are >10000 as Grade A, 5000-10000 as grade B and < 5000 as

Grade C, write an SQL query)

**#### Input :-**

```

+------+---------------+------+

|emp\_id|       emp\_name|salary|

+------+---------------+------+

|     1|           Jhon|  4000|

|     2|      Tim David| 12000|

|     3|Json Bhrendroff|  7000|

|     4|         Jordon|  8000|

|     5|          Green| 14000|

|     6|         Brewis|  6000|

+------+---------------+------+

```

**#### Expected Output :-**

```

+------+---------------+------+-----+

|emp\_id|       emp\_name|salary|grade|

+------+---------------+------+-----+

|     1|           Jhon|  4000|    C|

|     2|      Tim David| 12000|    A|

|     3|Json Bhrendroff|  7000|    B|

|     4|         Jordon|  8000|    B|

|     5|          Green| 14000|    A|

|     6|         Brewis|  6000|    B|

+------+---------------+------+-----+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio11.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio11.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**### Scenerio-12**

**#### Input :-**

```

+--------------------+----------+

|               email|    mobile|

+--------------------+----------+

|Renuka1992@gmail.com|9856765434|

|anbu.arasu@gmail.com|9844567788|

+--------------------+----------+

```

**#### Expected Output :-**

```

+--------------------+----------+

|               email|    mobile|

+--------------------+----------+

|R\*\*\*\*\*\*\*\*\*\*92@gma...|98\*\*\*\*\*434|

|a\*\*\*\*\*\*\*\*\*\*su@gma...|98\*\*\*\*\*788|

+--------------------+----------+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio12.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio12.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-13**

**####  (We have employee id,employee name, department. Need count of every department employees.)**

**#### Input :-**

```

+------+--------+-----------+

|emp\_id|emp\_name|       dept|

+------+--------+-----------+

|     1|    Jhon|Development|

|     2|     Tim|Development|

|     3|   David|    Testing|

|     4|     Sam|    Testing|

|     5|   Green|    Testing|

|     6|  Miller| Production|

|     7|  Brevis| Production|

|     8|  Warner| Production|

|     9|    Salt| Production|

+------+--------+-----------+

```

**#### Expected Output :-**

```

+-----------+-----+

|       dept|total|

+-----------+-----+

|Development|    2|

|    Testing|    3|

| Production|    4|

+-----------+-----+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio13.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio13.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-14**

**#### (We need total marks)**

**#### Input :-**

```

+------+------+------+-------+-----+-------+------+

|rollno|  name|telugu|english|maths|science|social|

+------+------+------+-------+-----+-------+------+

|203040|rajesh|    10|     20|   30|     40|    50|

+------+------+------+-------+-----+-------+------+

```

**#### Expected Output :-**

```

+------+------+------+-------+-----+-------+------+-----+

|rollno|  name|telugu|english|maths|science|social|total|

+------+------+------+-------+-----+-------+------+-----+

|203040|rajesh|    10|     20|   30|     40|    50|  150|

+------+------+------+-------+-----+-------+------+-----+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio14.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio14.py>) <br>

SQL -

```

select

  \*,

  (

    telugu + english + maths + science + social

  ) as total

from

  markstab

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-15**

**#### (Extend and Append list in  python and scala)**

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio15.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio15.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-16**

**#### (Remove duplicates)**

**#### Input :-**

```

+---+----+-----------+------+

| id|name|       dept|salary|

+---+----+-----------+------+

|  1|Jhon|    Testing|  5000|

|  2| Tim|Development|  6000|

|  3|Jhon|Development|  5000|

|  4| Sky| Prodcution|  8000|

+---+----+-----------+------+

```

**#### Expected Output :-**

```

+---+----+-----------+------+

| id|name|       dept|salary|

+---+----+-----------+------+

|  1|Jhon|    Testing|  5000|

|  2| Tim|Development|  6000|

|  4| Sky| Prodcution|  8000|

+---+----+-----------+------+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio16.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio16.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-17**

**#### (df1 contains Employeeid,Name,Age,State,Country columns df2 contains Employeeid,Name,Age,Address columns. how do you merge df1 and df2 to get the following output Employeeid,Name,Age,State,Country,Address)**

**#### Input :-**

```

+------+-----+---+------+-------+

|emp\_id| name|age| state|country|

+------+-----+---+------+-------+

|     1|  Tim| 24|Kerala|  India|

|     2|Asman| 26|Kerala|  India|

+------+-----+---+------+-------+

```

```

+------+-----+---+-------+

|emp\_id| name|age|address|

+------+-----+---+-------+

|     1|  Tim| 24|Comcity|

|     2|Asman| 26|bimcity|

+------+-----+---+-------+

```

**#### Expected Output :-**

```

+------+-----+---+------+-------+-------+

|emp\_id| name|age| state|country|address|

+------+-----+---+------+-------+-------+

|     1|  Tim| 24|Kerala|  India|Comcity|

|     2|Asman| 26|Kerala|  India|bimcity|

+------+-----+---+------+-------+-------+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio17.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio17.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-18**

**#### Input :-**

```

+------------------+

|              word|

+------------------+

|The Social Dilemma|

+------------------+

```

**#### Expected Output :-**

```

+------------------+

|      reverse word|

+------------------+

|ehT laicoS ammeliD|

+------------------+

```

**#### Solution :-**

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio18.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio18.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-19**

**#### (Flatten the below complex dataframe)**

**#### Input :-**

```

root

 |-- code: long (nullable = true)

 |-- commentCount: long (nullable = true)

 |-- createdAt: string (nullable = true)

 |-- description: string (nullable = true)

 |-- feedsComment: string (nullable = true)

 |-- id: long (nullable = true)

 |-- imagePaths: string (nullable = true)

 |-- images: string (nullable = true)

 |-- isdeleted: boolean (nullable = true)

 |-- lat: long (nullable = true)

 |-- likeDislike: struct (nullable = true)

 |    |-- dislikes: long (nullable = true)

 |    |-- likes: long (nullable = true)

 |    |-- userAction: long (nullable = true)

 |-- lng: long (nullable = true)

 |-- location: string (nullable = true)

 |-- mediatype: long (nullable = true)

 |-- msg: string (nullable = true)

 |-- multiMedia: array (nullable = true)

 |    |-- element: struct (containsNull = true)

 |    |    |-- createAt: string (nullable = true)

 |    |    |-- description: string (nullable = true)

 |    |    |-- id: long (nullable = true)

 |    |    |-- likeCount: long (nullable = true)

 |    |    |-- mediatype: long (nullable = true)

 |    |    |-- name: string (nullable = true)

 |    |    |-- place: string (nullable = true)

 |    |    |-- url: string (nullable = true)

 |-- name: string (nullable = true)

 |-- profilePicture: string (nullable = true)

 |-- title: string (nullable = true)

 |-- totalFeed: long (nullable = true)

 |-- userId: long (nullable = true)

 |-- videoUrl: string (nullable = true)

```

**#### Expected Output :-**

```

root

 |-- code: long (nullable = true)

 |-- commentCount: long (nullable = true)

 |-- createdAt: string (nullable = true)

 |-- description: string (nullable = true)

 |-- feedsComment: string (nullable = true)

 |-- id: long (nullable = true)

 |-- imagePaths: string (nullable = true)

 |-- images: string (nullable = true)

 |-- isdeleted: boolean (nullable = true)

 |-- lat: long (nullable = true)

 |-- lng: long (nullable = true)

 |-- location: string (nullable = true)

 |-- mediatype: long (nullable = true)

 |-- msg: string (nullable = true)

 |-- name: string (nullable = true)

 |-- profilePicture: string (nullable = true)

 |-- title: string (nullable = true)

 |-- totalFeed: long (nullable = true)

 |-- userId: long (nullable = true)

 |-- videoUrl: string (nullable = true)

 |-- dislikes: long (nullable = true)

 |-- likes: long (nullable = true)

 |-- userAction: long (nullable = true)

 |-- createAt: string (nullable = true)

 |-- likeCount: long (nullable = true)

 |-- place: string (nullable = true)

 |-- url: string (nullable = true)

```

**#### Solution :-**

Dataset - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Datasets/scen.json> <br>

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio19.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio19.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-20**

**#### (Generate the complex dataframe)**

**#### Input :-**

```

root

 |-- code: long (nullable = true)

 |-- commentCount: long (nullable = true)

 |-- createAt: string (nullable = true)

 |-- createdAt: string (nullable = true)

 |-- description: string (nullable = true)

 |-- dislikes: long (nullable = true)

 |-- feedsComment: string (nullable = true)

 |-- id: long (nullable = true)

 |-- imagePaths: string (nullable = true)

 |-- images: string (nullable = true)

 |-- isdeleted: boolean (nullable = true)

 |-- lat: long (nullable = true)

 |-- likeCount: long (nullable = true)

 |-- likes: long (nullable = true)

 |-- lng: long (nullable = true)

 |-- location: string (nullable = true)

 |-- mediatype: long (nullable = true)

 |-- msg: string (nullable = true)

 |-- name: string (nullable = true)

 |-- place: string (nullable = true)

 |-- profilePicture: string (nullable = true)

 |-- title: string (nullable = true)

 |-- totalFeed: long (nullable = true)

 |-- url: string (nullable = true)

 |-- userAction: long (nullable = true)

 |-- userId: long (nullable = true)

 |-- videoUrl: string (nullable = true)

```

**#### Expected Output :-**

```

root

 |-- code: long (nullable = true)

 |-- commentCount: long (nullable = true)

 |-- createdAt: string (nullable = true)

 |-- description: string (nullable = true)

 |-- feedsComment: string (nullable = true)

 |-- id: long (nullable = true)

 |-- imagePaths: string (nullable = true)

 |-- images: string (nullable = true)

 |-- isdeleted: boolean (nullable = true)

 |-- lat: long (nullable = true)

 |-- likeDislike: struct (nullable = false)

 |    |-- dislikes: long (nullable = true)

 |    |-- likes: long (nullable = true)

 |    |-- userAction: long (nullable = true)

 |-- lng: long (nullable = true)

 |-- location: string (nullable = true)

 |-- mediatype: long (nullable = true)

 |-- msg: string (nullable = true)

 |-- multiMedia: array (nullable = false)

 |    |-- element: struct (containsNull = false)

 |    |    |-- createAt: string (nullable = true)

 |    |    |-- description: string (nullable = true)

 |    |    |-- id: long (nullable = true)

 |    |    |-- likeCount: long (nullable = true)

 |    |    |-- mediatype: long (nullable = true)

 |    |    |-- name: string (nullable = true)

 |    |    |-- place: string (nullable = true)

 |    |    |-- url: string (nullable = true)

 |-- name: string (nullable = true)

 |-- profilePicture: string (nullable = true)

 |-- title: string (nullable = true)

 |-- userId: long (nullable = true)

 |-- videoUrl: string (nullable = true)

 |-- totalFeed: long (nullable = true)

```

**#### Solution :-**

Dataset - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Datasets/scen20.json> <br>

Scala-Spark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio20.scala> <br>

PySpark - <https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio20.py>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-21**

**#### (The roundtrip distance should be calculated using spark or SQL.)**

**#### Input :-**

```

+----+---+----+

|from| to|dist|

+----+---+----+

| SEA| SF| 300|

| CHI|SEA|2000|

|  SF|SEA| 300|

| SEA|CHI|2000|

| SEA|LND| 500|

| LND|SEA| 500|

| LND|CHI|1000|

| CHI|NDL| 180|

+----+---+----+

```

**#### Expected Output :-**

```

+----+---+--------------+

|from| to|roundtrip\_dist|

+----+---+--------------+

| SEA| SF|           600|

| CHI|SEA|          4000|

| LND|SEA|          1000|

+----+---+--------------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio21.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio21.py>) <br>

SQL -

```

select

  r1.from,

  r1.to,

  (r1.dist + r2.dist) as round\_distance

from

  trip r1

  join trip r2 on r1.from = r2.to

  and r1.to = r2.from

where

  r1.from < r1.to

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-22**

**#### (Cumilative sum)**

**#### Input :-**

```

+---+------+-----+

|pid|  date|price|

+---+------+-----+

|  1|26-May|  100|

|  1|27-May|  200|

|  1|28-May|  300|

|  2|29-May|  400|

|  3|30-May|  500|

|  3|31-May|  600|

+---+------+-----+

```

**#### Expected Output :-**

```

+---+------+-----+---------+

|pid|  date|price|new\_price|

+---+------+-----+---------+

|  1|26-May|  100|      100|

|  1|27-May|  200|      300|

|  1|28-May|  300|      600|

|  2|29-May|  400|      400|

|  3|30-May|  500|      500|

|  3|31-May|  600|     1100|

+---+------+-----+---------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio22.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio22.py>) <br>

SQL -

```

select

  pid,

  date,

  price,

  sum(price) over (

    partition by pid

    order by

      price

  ) as newprice

from

  ordertab

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-23**

**#### Input :-**

```

+-----------+-----------+

|customer\_id|product\_key|

+-----------+-----------+

|          1|          5|

|          2|          6|

|          3|          5|

|          3|          6|

|          1|          6|

+-----------+-----------+

```

```

+-----------+

|product\_key|

+-----------+

|          5|

|          6|

+-----------+

```

**#### Expected Output :-**

```

+-----------+

|customer\_id|

+-----------+

|          1|

|          3|

+-----------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio23.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio23.py>)

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-24**

**#### Input :-**

```

+------+------------+

|userid|        page|

+------+------------+

|     1|        home|

|     1|    products|

|     1|    checkout|

|     1|confirmation|

|     2|        home|

|     2|    products|

|     2|        cart|

|     2|    checkout|

|     2|confirmation|

|     2|        home|

|     2|    products|

+------+------------+

```

**#### Expected Output :-**

```

+------+--------------------------------------------------------------+

|userid|pages                                                         |

+------+--------------------------------------------------------------+

|1     |[home, products, checkout, confirmation]                      |

|2     |[home, products, cart, checkout, confirmation, home, products]|

+------+--------------------------------------------------------------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio24.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio24.py>) <br>

SQL :-

```

select

  userid,

  collect\_list(page) as pages

from

  testcol

group by

  userid;

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-25**

**### consider a file with some bad/corrupt data as shown below.How will you handle those and load into spark dataframe**

Note - avoid using filter after reading as DF and try to remove bad data while reading the file itself

**#### Input :-**

```

emp\_no,emp\_name,dep

101,Murugan,HealthCare

Invalid Entry,Description: Bad Record Entry

102,Kannan,Finance

103,Mani,IT

Connection lost,Description: Poor Connection

104,Pavan,HR

Bad Record,Description:Corrupt Record

```

**#### Expected Output :-**

```

+------+--------+----------+

|emp\_no|emp\_name|       dep|

+------+--------+----------+

|   101| Murugan|HealthCare|

|   102|  Kannan|   Finance|

|   103|    Mani|        IT|

|   104|   Pavan|        HR|

+------+--------+----------+

```

**#### Solution :-**

Scala-Spark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio25.scala>) <br>

PySpark - [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio25.py>)

There are three modes available when reading a file in Spark:

\* `PERMISSIVE` : This is the default mode. It attempts to parse all the rows in the file, and if it encounters any malformed data or parsing errors, it sets the problematic fields to null and adds a new column called \_corrupt\_record to store the entire problematic row as a string.

\* `DROPMALFORMED` : This mode drops the rows that contain malformed data or cannot be parsed according to the specified schema. It only includes the rows that can be successfully parsed.

\* `FAILFAST` : This mode throws an exception and fails immediately if it encounters any malformed data or parsing errors in the file. It does not process any further rows after the first encountered error.

You can specify the desired mode using the mode option when reading a file, such as option("mode", "PERMISSIVE") or option("mode", "FAILFAST"). If the mode option is not explicitly set, it defaults to PERMISSIVE.

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-26**

\* Input :-

```sh

+---+----+

| id|name|

+---+----+

|  1|   A|

|  2|   B|

|  3|   C|

|  4|   D|

+---+----+

+---+-----+

|id1|name1|

+---+-----+

|  1|    A|

|  2|    B|

|  4|    X|

|  5|    F|

+---+-----+

```

\* Output :-

```sh

+---+-------------+

| id|      comment|

+---+-------------+

|  3|new in source|

|  4|     mismatch|

|  5|new in target|

+---+-------------+

```

**#### Solution :-**

Scala-Spark :- [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio26.scala>) <br>

PySpark :- [Click Here](<https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio26.py>) <br>

SQL :-

```

select

  id,

  case when name != name1 then 'Mismatch' when name1 is null then 'New in Source' when name is null then 'New in Target' end as comment

from

  (

    select

      coalesce(id, id1) as id,

      s.name,

      t.name1

    from

      sourcetab s full

      outer join targettab t on s.id = t.id1

    WHERE

      s.name != t.name1

      OR s.name IS NULL

      OR t.name1 IS NULL

  );

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-27**

\* Input :-

```sh

+-----+------+----+

|empid|salary|year|

+-----+------+----+

|    1| 60000|2018|

|    1| 70000|2019|

|    1| 80000|2020|

|    2| 60000|2018|

|    2| 65000|2019|

|    2| 65000|2020|

|    3| 60000|2018|

|    3| 65000|2019|

+-----+------+----+

```

\* Output :-

```sh

+-----+------+----+-----------+

|empid|salary|year|incresalary|

+-----+------+----+-----------+

|    1| 60000|2018|          0|

|    1| 70000|2019|      10000|

|    1| 80000|2020|      10000|

|    2| 60000|2018|          0|

|    2| 65000|2019|       5000|

|    2| 65000|2020|          0|

|    3| 60000|2018|          0|

|    3| 65000|2019|       5000|

+-----+------+----+-----------+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio27.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio27.py) <br>

SQL :-

```

select

  empid,

  salary,

  year,

  coalesce(

    (salary - diff),

    0

  ) as increment

from

  (

    select

      \*,

      lag(salary, 1) over (

        partition by empid

        order by

          year

      ) as diff

    from

      salarytab

  );

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-28**

\* Input :-

```sh

+-----+------+

|child|parent|

+-----+------+

|    A|    AA|

|    B|    BB|

|    C|    CC|

|   AA|   AAA|

|   BB|   BBB|

|   CC|   CCC|

+-----+------+

```

\* Output :-

```sh

+-----+------+-----------+

|child|parent|grandparent|

+-----+------+-----------+

|    A|    AA|        AAA|

|    C|    CC|        CCC|

|    B|    BB|        BBB|

+-----+------+-----------+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio28.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio28.py)

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-29**

\* Input :-

```sh

+---+

|col|

+---+

|  1|

|  2|

|  3|

+---+

+----+

|col1|

+----+

|   1|

|   2|

|   3|

|   4|

|   5|

+----+

```

\* Output :-

```sh

+---+

|col|

+---+

|  1|

|  2|

|  4|

|  5|

+---+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio29.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio29.py)

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-30**

\* Write a SQL Query to extract second most salary for each department

\* Input :-

```sh

+------+----+-------+-------+

|emp\_id|name|dept\_id| salary|

+------+----+-------+-------+

|     1|   A|      A|1000000|

|     2|   B|      A|2500000|

|     3|   C|      G| 500000|

|     4|   D|      G| 800000|

|     5|   E|      W|9000000|

|     6|   F|      W|2000000|

+------+----+-------+-------+

+--------+---------+

|dept\_id1|dept\_name|

+--------+---------+

|       A|    AZURE|

|       G|      GCP|

|       W|      AWS|

+--------+---------+

```

\* Output :-

```sh

+------+----+---------+-------+

|emp\_id|name|dept\_name| salary|

+------+----+---------+-------+

|     1|   A|    AZURE|1000000|

|     6|   F|      AWS|2000000|

|     3|   C|      GCP| 500000|

+------+----+---------+-------+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio30.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio30.ipynb) <br>

SQL :-

```sh

WITH jointab AS (

    SELECT df1.emp\_id, df1.name, df1.dept\_id, df1.salary, df2.dept\_name,

           DENSE\_RANK() OVER (PARTITION BY df1.dept\_id ORDER BY df1.salary DESC) AS row\_rank

    FROM df1

    INNER JOIN df2 ON df1.dept\_id = df2.dept\_id1

)

SELECT emp\_id,name,dept\_name,salary from jointab WHERE row\_rank =2;

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-31**

\* Input :-

```sh

+----+-----+--------+-----------+

|col1| col2|    col3|       col4|

+----+-----+--------+-----------+

|  m1|m1,m2|m1,m2,m3|m1,m2,m3,m4|

+----+-----+--------+-----------+

```

\* Output :-

```sh

+-----------+

|        col|

+-----------+

|         m1|

|      m1,m2|

|   m1,m2,m3|

|m1,m2,m3,m4|

|           |

+-----------+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio31.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio31.ipynb) <br>

SQL :-

```sh

select

  explode(

    split(col, '-')

  )

from

  (

    select

      concat(

        col1, '-', col2, '-', col3, '-', col4

      ) as col

    from

      mtab

  );

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-32**

\* Input :-

```sh

+-------+-------------------+

|food\_id|          food\_item|

+-------+-------------------+

|      1|        Veg Biryani|

|      2|     Veg Fried Rice|

|      3|    Kaju Fried Rice|

|      4|    Chicken Biryani|

|      5|Chicken Dum Biryani|

|      6|     Prawns Biryani|

|      7|      Fish Birayani|

+-------+-------------------+

+-------+------+

|food\_id|rating|

+-------+------+

|      1|     5|

|      2|     3|

|      3|     4|

|      4|     4|

|      5|     5|

|      6|     4|

|      7|     4|

+-------+------+

```

\* Output :-

```sh

+-------+-------------------+------+---------------+

|food\_id|          food\_item|rating|stats(out of 5)|

+-------+-------------------+------+---------------+

|      1|        Veg Biryani|     5|          \*\*\*\*\*|

|      2|     Veg Fried Rice|     3|            \*\*\*|

|      3|    Kaju Fried Rice|     4|           \*\*\*\*|

|      4|    Chicken Biryani|     4|           \*\*\*\*|

|      5|Chicken Dum Biryani|     5|          \*\*\*\*\*|

|      6|     Prawns Biryani|     4|           \*\*\*\*|

|      7|      Fish Birayani|     4|           \*\*\*\*|

+-------+-------------------+------+---------------+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio32%20Scala.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio32.ipynb) <br>

SQL :-

```sh

select

  foodtab.food\_id,

  foodtab.food\_item,

  ratingtab.rating,

  repeat('\*', ratingtab.rating) as stars

from

  foodtab

  inner join ratingtab on foodtab.food\_id = ratingtab.food\_id

order by

  foodtab.food\_id;

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-33**

\* Write a query to print the maximum number of discount tours any 1 family can choose.

\* Input :-

```sh

+--------------------+--------------+-----------+

|                  id|          name|family\_size|

+--------------------+--------------+-----------+

|c00dac11bde74750b...|   Alex Thomas|          9|

|eb6f2d3426694667a...|    Chris Gray|          2|

|3f7b5b8e835d4e1c8...| Emily Johnson|          4|

|9a345b079d9f4d3ca...| Michael Brown|          6|

|e0a5f57516024de2a...|Jessica Wilson|          3|

+--------------------+--------------+-----------+

+--------------------+------------+--------+--------+

|                  id|        name|min\_size|max\_size|

+--------------------+------------+--------+--------+

|023fd23615bd4ff4b...|     Bolivia|       2|       4|

|be247f73de0f4b2d8...|Cook Islands|       4|       8|

|3e85ab80a6f84ef3b...|      Brazil|       4|       7|

|e571e164152c4f7c8...|   Australia|       5|       9|

|f35a7bb7d44342f7a...|      Canada|       3|       5|

|a1b5a4b5fc5f46f89...|       Japan|      10|      12|

+--------------------+------------+--------+--------+

```

\* Output :-

```sh

+-------------+-------------------+

|         name|number\_of\_countries|

+-------------+-------------------+

|Emily Johnson|                  4|

+-------------+-------------------+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio33.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio33.ipynb) <br>

SQL :-

```sh

select max(number\_of\_countries) from (select f.name,count(\*) as number\_of\_countries from family f inner join country c on f.family\_size  between c.min\_size and c.max\_size group by f.name);

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-34**

\* Input :-

```sh

+-----------+------+---+------+

|customer\_id|  name|age|gender|

+-----------+------+---+------+

|          1| Alice| 25|     F|

|          2|   Bob| 40|     M|

|          3|   Raj| 46|     M|

|          4| Sekar| 66|     M|

|          5|  Jhon| 47|     M|

|          6|Timoty| 28|     M|

|          7|  Brad| 90|     M|

|          8|  Rita| 34|     F|

+-----------+------+---+------+

```

\* Output :-

```sh

+---------+-----+

|age\_group|count|

+---------+-----+

|    19-35|    3|

|    36-50|    3|

|      51+|    2|

+---------+-----+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio34.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio34.ipynb) <br>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-35**

Question (IBM Question)

\* Create a new datafrane df1 with the given values

\* Count null entries in a datafarme

\* Remove null entries and the store the null entries in a new datafarme df2

\* Create a new dataframe df3 with the given values and join the two dataframes df1 & df2

\* Fill the null values with the mean age all of students

\* Filter the students who are 18 years above and older

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio35.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio35.ipynb) <br>

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***

**## Scenerio-36**

\* Input :-

```sh

+----------+----------+

| sell\_date|   product|

+----------+----------+

|2020-05-30| Headphone|

|2020-06-01|    Pencil|

|2020-06-02|      Mask|

|2020-05-30|Basketball|

|2020-06-01|      Book|

|2020-06-02|      Mask|

|2020-05-30|   T-Shirt|

+----------+----------+

```

\* Output :-

```sh

+----------+--------------------+---------+

| sell\_date|            products|null\_sell|

+----------+--------------------+---------+

|2020-05-30|[T-Shirt, Basketb...|        3|

|2020-06-01|      [Pencil, Book]|        2|

|2020-06-02|              [Mask]|        1|

+----------+--------------------+---------+

```

**#### Solution :-**

Scala-Spark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/src/pack/Scenerio36.scala) <br>

PySpark :- [Click Here](https://github.com/mohankrishna02/interview-scenerios-spark-sql/blob/master/Scenerio36.ipynb) <br>

SQL :-

```sh

select sell\_date,(collect\_set(product)) as products,size(collect\_set(product)) as num\_sell from products group by sell\_date;

```

**\*\*[⬆ Back to Top](**#table-of-contents**)\*\***