

Day39 - March 15th 2024

1. Started my day as usual
2. Completed one leetcode problem and will be consistently solved problem her on
3. Learning spark fundamentals from manish kumar

The screenshot shows a dual-monitor setup. The left monitor displays a YouTube video titled "spark fundamental (Theory)" by Manish Kumar. The video content includes handwritten notes: "Features of an RDD?", "what is dataframe / dataset?", "why we should not use an RDD?", and "RDD -> Resilient". The right monitor displays a Google Docs document titled "Spark Theory". It contains text about "Logical Optimization" and a diagram showing the flow from "Logical plan" to "Optimized logical plan" to "Physical plan" to "Cost model" to "Best Physical plan" to "Final Code RDD's". The diagram also includes "RDD -> cluster" and "RDD -> Resilient".

4. More details about the docs :
https://docs.google.com/document/d/1vtgtCNflkiyHrPocS1pdTSaT2FpQFQBikt_0ljkNISc/edit?usp=sharing
5. Ended my day my solving a complex SQL question from Ankit's Complex playlist

SQLQuery1.sql - KAUSHI\SQLEXPRESS.master (KAUSHI\iamka (69)) - Microsoft SQL Server Management Studio

```

--select p1.userid from purchase_history p1
--left join purchase_history p2 on p1.userid = p2.userid and p2.productid > p1.productid and p1.purchasedate != p2.purchasedate
--where p2.userid is null and p2.productid is null
--group by p1.userid
--having count(p1.productid) = 1

--select p1.*,p2.* from purchase_history p1
--left join purchase_history p2 on p1.userid = p2.userid and p2.productid > p1.productid and p1.purchasedate != p2.purchasedate

--select userid,purchasedate,count(productid)
--from purchase_history
--group by userid,purchasedate

```

88 %

Results Messages

	userid	purchasedate	count
1	1	2012-01-23	2
2	2	2012-01-23	2
3	3	2012-01-23	2
4	4	2012-01-23	1
5	1	2012-01-25	1
6	2	2012-01-25	2
7	4	2012-01-25	1

Query executed successfully.

KAUSHI\SQLEXPRESS (16.0 RTM) KAUSHI\iamka (69) master 00:00:00 7 rows

Ready 10°C Clear

6.

SQLQuery1.sql - KAUSHI\SQLEXPRESS.master (KAUSHI\iamka (69)) - Microsoft SQL Server Management Studio

```

--with cte as(
--select *,max(productid) over(partition by userid,purchasedate order by userid) as m
--from purchase_history)
--select c1.*,c2.* from cte c1
--left join cte c2 on c1.userid = c2.userid and c2.productid > c1.productid and c1.purchasedate != c2.purchasedate

--select p1.userid from purchase_history p1
--left join purchase_history p2 on p1.userid = p2.userid and p2.productid > p1.productid and p1.purchasedate != p2.purchasedate
--where p2.userid is null and p2.productid is null
--group by p1.userid
--having count(p1.productid) = 1

--select p1.*,p2.* from purchase_history p1
--left join purchase_history p2 on p1.userid = p2.userid and p2.productid > p1.productid and p1.purchasedate != p2.purchasedate

--select userid,purchasedate,count(productid)
--from purchase_history
--group by userid,purchasedate

--select * from purchase_history;

--with cte as(
--select *,max(productid) over(partition by userid,purchasedate order by userid) as m
--from purchase_history)
--select c1.*,c2.* from cte c1
--inner join cte c2 on c1.purchasedate < c2.purchasedate and c1.userid = c2.userid

--with cte as(
--select *,max(productid) over(partition by userid,purchasedate order by userid) as m
--from purchase_history)
--select c1.*,c2.* from cte c1
--left join cte c2 on c1.purchasedate < c2.purchasedate and c1.userid=c2.userid

--select userid,purchasedate,count(distinct productid)
--from purchase_history
--group by userid,purchasedate
--order by userid

--select * from purchase_history

```

88 %

Results Messages

	userid
1	1
2	4

Query executed successfully.

KAUSHI\SQLEXPRESS (16.0 RTM) KAUSHI\iamka (69) master 00:00:00 2 rows

Ready 9°C Haze

7. Wasted lot of time by thinking for a efficient solution

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
-- Create table purchase_history
create table purchase_history
(
    userid int,
    productid int,
    purchasedate date
);
SET DATEFORMAT day;
-- Insert into purchase_history values
insert into purchase_history values
(1,1,'23-01-2012'),
(1,2,'23-01-2012'),
(1,3,'25-01-2012'),
(2,1,'23-01-2012'),
(2,2,'23-01-2012'),
(2,2,'25-01-2012'),
(2,4,'25-01-2012'),
(3,4,'23-01-2012'),
(3,1,'23-01-2012'),
(4,1,'23-01-2012'),
(4,2,'25-01-2012');
-- Q : Write a SQL query to find user's who purchased different products on different dates
select * from purchase_history
```

The Results pane shows the following data:

userid	productid	purchasedate
1	1	2012-01-23
1	2	2012-01-23
1	3	2012-01-25
2	1	2012-01-23
2	2	2012-01-23
2	2	2012-01-25
2	4	2012-01-25
3	4	2012-01-23
3	1	2012-01-23
4	1	2012-01-23
4	2	2012-01-25

The status bar at the bottom indicates "Query executed successfully." and "11 rows".

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
-- select p1.userid from purchase_history p1
-- left join purchase_history p2 on p1.userid = p2.userid and p1.productid != p2.productid
-- and p1.purchasedate < p2.purchasedate
-- group by p1.userid
-- having count(p1.userid) - 1 = count(p2.userid)

/* Explanation:
Step1 : First I have used self left join on p1.userid = p2.userid and p1.productid != p2.productid
and p1.purchasedate < p2.purchasedate
Step2 : The self left join lets us compare the product_id from two different purchasedates
Step3 : Now from here we'll select the userid and count(p1.productid)-1 = count(p2.productid)
If count(p1.productid)-1 = count(p2.productid) satisfies then it tells us that userid
has ordered different products on different dates
*/

/* but ankit solved this question in most efficient way
sol : select userid from purchasehistory
group by userid
having count(purchasedate)>1 and count(productid) = count(distinct productid)
*/
```

The Results pane shows the following data:

userid
1
4

The status bar at the bottom indicates "Query executed successfully." and "2 rows".