BUT IN REALITY, QUERIES ARE PRETTY PLUG-AND-PLAY

WE COULD USE ONE QUERY INSIDE ANOTHER (VIA SUBQUERIES)

WE COULD CALCULATE THE UNION, INTERSECTION OR DIFFERENCE OF 2 QUERIES

WE COULD USE A SUBQUERY TO POPULATE A TABLE VIA INSERT

WE COULD USE ONE QUERY INSIDE ANOTHER (VIA SUBQUERIES)

'STORES'

StoreID	StoreLocation	City
1	Bellandur	Bangalore
2	Koramangala	Bangalore

'PRODUCTS'

ProductID	ProductName
1	Bananas
2	Milk
3	Nutella
4	Peanut Butter

'SALES_DATA'

StoreID	ProductID	Date	Revenue
1	1	January 18,2016	8,236.33
1	3	January 18,2016	7,455.67
1	4	January 18,2016	5,316.89
1	2	January 18,2016	2,433.76
2	1	January 18,2016	9,456.01
2	3	January 18,2016	3,644.33
2	4	January 18,2016	8,988.64

WE COULD USE ONE QUERY INSIDE ANOTHER (VIA SUBQUERIES)

IF SUCH A DATABASE ACTUALLY EXISTED

WE COULD USE ONE
QUERY INSIDE ANOTHER
(VIA SUBQUERIES)

IF SUCH A DATABASE ACTUALLY EXISTED

YOU CAN BET THAT SOMEONE
WOULD PULL A REPORT FOR
ANNUAL REVENUE

WE COULD USE ONE
QUERY INSIDE ANOTHER
(VIA SUBQUERIES)

IF SUCH A DATABASE ACTUALLY EXISTED

YOU CAN BET THAT SOMEONE WOULD
PULL A REPORT FOR ANNUAL REVENUE

IN FACT, THERE WOULD BE A
REALLY BORING WEEKLY MEETING
AROUND THIS REPORT

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.ProductID = p.ProductID
WHERE
    (p.ProductName = 'Peanut Butter' or p.ProductName =
    'Nutella') AND (YEAR(date) = 2016)
GROUP BY
     p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.ProductID = p.ProductID
WHERE
     (p.ProductName = 'Peanut Butter' or p.ProductName =
     'Nutella') AND (YEAR(date) = 2016)
GROUP BY
     p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.ProductID = p.ProductID
WHERE
     (p.ProductName = 'Peanut Butter' or p.ProductName =
     'Nutella') AND (YEAR(date) = 2016)
GROUP BY
     p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
    p.ProductName,YEAR(date),SUM(revenue)
FROM
    Sales_Data s
INNER JOIN
    Products p
    ON
    s.ProductID = p.ProductID
WHERE
    (p.ProductName = 'Peanut Butter' or p.ProductName =
    'Nutella') AND (YEAR(date) = 2016)
GROUP BY
    p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
    Sales_Data s
INNER JOIN
    Products p
    ON
    s.ProductID = p.ProductID
WHERE
    (p.ProductName = 'Peanut Butter' or p.ProductName =
    'Nutella') AND (YEAR(date) = 2016)
GROUP BY
    p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.
WHERE
     (p.
     'N
GROUP
     p.
```

ERRM..THIS QUERY WILL NEED TO BE RE-WRITTEN EACH YEAR :-|

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.
WHERE
     (p.
     'Nu
GROUP
     p.
```

NOT IF WE USE A SUB-QUERY!

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.
WHERE
     (p.
     'Nu
GROUP
     p.
```

NOT IF WE USE A SUB-QUERY!

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'DATA'

Revenue

WE JUST PLUGGED ONE QUERY INTO ANOTHER!

```
SELECT
    p.ProductName, YEAR(date), SUM(s.revenue)
FROM
    S
INNER
    P
    O
    s
WHERE
    (p.ProductName = 'Peanut Butter' or p.ProductName =
    'Nutella') AND
    (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data))
GROUP BY
    p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

```
SELECT
      p.ProductName,YEAR(date),SUM(revenue)
FROM

INNE

WHER
      'Nutella') AND
      (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data))
GROUP BY
      p.ProductName, YEAR(date)
```

_DATA'

Revenue

HERE, THE INNER QUERY RETURNS A SINGLE VALUE..

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

```
SELECT
      P.ProductName,YEAR(date),SUM(revenue)
FRO
INN
WHE
      'Nutella') AND
      (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data))
GROUP BY
      P.ProductName, YEAR(date)
```

_DATA'

Revenue

AND THE OUTER QUERY USES THAT VALUE IN THE WHERE CLAUSE

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.ProductID = p.ProductID
WHERE
    (p.ProductName = 'Peanut Butter' or p.ProductName =
    'Nutella') AND
    (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data))
GROUP BY
     p.ProductName, YEAR(date)
```

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

YOU CAN BET THAT

A

LET'S SAY WE DON'T WANT TO HARD CODE THE PRODUCTS, INSTEAD WE HAVE A TOP SELLERS TABLE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES_DATA'

eID	ProductID	Date	Revenue

```
    ON
    s.ProductID = p.ProductID
WHERE
    (p.ProductName = 'Peanut Butter' or p.ProductName =
    'Nutella') AND
    (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data))
GROUP BY
    p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

ALES_DATA'

ate	Revenue

WE JUST PLUGGED ONE QUERY INTO ANOTHER!

```
SELECT
WHERE
    (p.ProductName IN (SELECT ProductName FROM Top_Sellers))
    AND
    (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data))
GROUP BY
    p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

ALES_DATA'

ate	Revenue

HERE, THE INNER QUERY RETURNS A RANGE OF VALUES..

```
SELECT
WHERE
    (P.ProductName IN (SELECT ProductName FROM Top_Sellers))
    AND
    (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data)
GROUP BY
    P.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'ALES_DATA'

ate	Revenue

AND THE OUTER QUERY USES ALL OF THOSE VALUES IN THE WHERE CLAUSE

```
SELECT
WHERE
    (P.ProductName IN (SELECT ProductName FROM Top_Sellers))
    AND
    (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data)
GROUP BY
    P.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
     p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.ProductID = p.ProductID
WHERE
    (p.ProductName IN (SELECT ProductName FROM Top_Sellers))
    AND
    (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data))
GROUP BY
     p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

```
SELECT
      p.ProductName,YEAR(date),SUM(revenue)
FROM
     Sales_Data s
INNER JOIN
     Products p
     ON
     s.
WHERE
     (p.
    AND
     (YE
GROUP
     p.
```

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

TIME PASSES, AND OUR SALES_DATA TABLE BECOMES ENORMOUS - TOO BIG TO PULL LIKE THIS

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES DATA'

	evenue

```
SELECT
      p.ProductName,YEAR(date),SUM(revenue)
FROM
      (SELECT * FROM Sales_Data WHERE ProductID IN(SELECT
       ProductID FROM Top_Sellers)) AS s
INNER J
      Pr
      ON
      s.
WHERE
     (p.
     (YE
GROUP B
     p.ProductName, YEAR(date)
```

WE JUST PLUGGED ONE QUERY INTO ANOTHER!

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES_DATA'

evenue

```
SELECT
      p.ProductName,YEAR(date),SUM(revenue)
FROM
      (SELECT * FROM Sales_Data WHERE ProductID IN(SELECT
       ProductID FROM Top_Sellers)) AS s
INNER J
      Pr
      ON
      s.
WHERE
     (p.
     (YE
GROUP B
     p.ProductName, YEAR(date)
```

HERE, THE INNER QUERY IS A SUBSET OF A TABLE...

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES_DATA'

evenue

```
SELECT
      p.ProductName,YEAR(date),SUM(revenue)
FROM
      (SELECT * FROM Sales_Data WHERE ProductID IN(SELECT
       ProductID FROM Top_Sellers)) AS s
INNER J
      Pr
      ON
      s.
WHERE
     (p.
     (YE
GROUP B
     p.ProductName, YEAR(date)
```

THE OUTER QUERY USES THAT TABLE SUBSET (ALSO A TABLE) IN THE FROM CLAUSE

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES_DATA'

StoreID	ProductID	Date	Revenue

```
SELECT
      p.ProductName,YEAR(date),SUM(revenue)
FROM
      (SELECT * FROM Sales_Data WHERE ProductID IN(SELECT
       ProductID FROM Top_Sellers)) AS s
INNER JOIN
     Products p
     ON
     s.ProductID = p.ProductID
WHERE
    (p.ProductName IN (SELECT ProductName FROM Top_Sellers)) AND
    (YEAR(date) = (SELECT YEAR(MAX(date)) FROM Sales_Data))
GROUP BY
    p.ProductName, YEAR(date)
```

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

'PRODUCTS'

ProductID	ProductName

'TOP SELLERS'

ProductID	ProductName

'SALES_DATA'

	venue

```
SELECT
      p.ProductName,YEAR(date),SUM(revenue)
FROM
      (SELECT * FROM Sales_Data WHERE ProductID IN(SELECT
       ProductID FROM Top_Sellers)) AS s
INNER JOIN
     Pro
     ON
     s.P
WHERE
     (p.P
     (YEA
GROUP B
     p.ProductName, YEAR(date)
```

THE QUERY IS NOW ENTIRELY FREE OF HARDCODED VALUES OR VERY LARGE INTERMEDIATE TABLES - THANKS TO THE USE OF SUBQUERIES!

WE COULD USE ONE QUERY INSIDE ANOTHER (VIA SUBQUERIES)

IF SUCH A DATABASE ACTUALLY EXISTED

YOU CAN BET THAT SOMEONE WOULD PULL A REPORT FOR ANNUAL REVENUE

IN FACT, THERE WOULD BE A REALLY BORING WEEKLY MEETING AROUND THIS REPORT

THE QUERY IS NOW ENTIRELY FREE OF HARDCODED VALUES OR VERY LARGE INTERMEDIATE TABLES - THANKS TO THE USE OF SUBQUERIES!

WE COULD USE ONE QUERY INSIDE ANOTHER (VIA SUBQUERIES)

'STORES'

StoreID	StoreLocation	City
1	Bellandur	Bangalore
2	Koramangala	Bangalore

'PRODUCTS'

ProductID	ProductName
1	Bananas
2	Milk
3	Nutella
4	Peanut Butter

SUBQUERIES REALLY SIMPLIFY & ROBUSTIFY QUERIES :-)

'SALES_DATA'

StoreID	ProductID	Date	Revenue
1	1	January 18,2016	8,236.33
1	3	January 18,2016	7,455.67
1	4	January 18,2016	5,216.89

WE COULD USE ONE
QUERY INSIDE ANOTHER
(VIA SUBQUERIES)

SUBQUERIES REALLY SIMPLIFY &
ROBUSTLY QUERIES :-)

BUT IN REALITY, QUERIES ARE PRETTY PLUG-AND-PLAY

WE COULD USE ONE QUERY INSIDE ANOTHER (VIA SUBQUERIES)

WE COULD CALCULATE THE UNION, INTERSECTION OR DIFFERENCE OF 2 QUERIES

WE COULD USE A SUBQUERY TO POPULATE A TABLE VIA INSERT

BUT IN REALITY, QUERIES ARE PRETTY PLUG-AND-PLAY

WE COULD USE ONE QUERY INSIDE ANOTHER (VIA SUBQUERIES)

WE COULD CALCULATE THE UNION, INTERSECTION OR DIFFERENCE OF 2 QUERIES

WE COULD USE A SUBQUERY TO POPULATE A TABLE VIA INSERT