

CREATING TABLES WITH PARTITIONS

CREATING TABLES WITH PARTITIONS

HIVE ORGANIZES TABLES INTO PARTITIONS

PARTITIONS ARE JUST SPLITS OF ALL THE DATA IN A TABLE

CREATING TABLES WITH PARTITIONS

HIVE ORGANIZES TABLES
INTO PARTITIONS

EACH PARTITION IS IN A SEPARATE
DIRECTORY

CREATING TABLES WITH PARTITIONS

HIVE ORGANIZES TABLES
INTO PARTITIONS

EACH PARTITION IS IN A SEPARATE
DIRECTORY

A DIRECTORY CAN HAVE MULTIPLE FILES
WITH THE DATA IN THAT PARTITION

CREATING TABLES WITH PARTITIONS

HIVE ORGANIZES TABLES
INTO PARTITIONS

ALL PARTITIONS TOGETHER MAKE UP
THE ENTIRE TABLE

CREATING TABLES WITH PARTITIONS

HIVE ORGANIZES TABLES
INTO PARTITIONS

THE PARTITION CAN BE BASED ON THE
VALUES OF ONE OR MORE COLUMNS
(CALLED PARTITION COLUMNS)

HIVE ORGANIZES TABLES INTO PARTITIONS
THE PARTITION CAN BE BASED ON THE VALUES OF
ONE OR MORE COLUMNS
(CALLED PARTITION COLUMNS)

YOU CAN USE ANY COLUMN TO CREATE PARTITIONS

StoreLocation	Product	Date	Revenue
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
Koramangala	Nutella	January 18,2016	3,644.33
Koramangala	Peanut Butter	January 18,2016	8,988.64
Koramangala	Milk	January 18,2016	1,621.58

CONSIDER THE SALES TABLE

YOU CAN USE ANY COLUMN TO CREATE PARTITIONS

StoreLocation	Product	Date	Revenue
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
Koramangala	Nutella	January 18,2016	3,644.33
Koramangala	Peanut Butter	January 18,2016	8,988.64
Koramangala	Milk	January 18,2016	1,621.58

CONSIDER THE SALES TABLE

LET'S PARTITION IT ON THE PRODUCT
COLUMN

YOU CAN USE ANY COLUMN TO CREATE PARTITIONS

StoreLocation	Product	Date	Revenue
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
Koramangala	Nutella	January 18,2016	3,644.33
Koramangala	Peanut Butter	January 18,2016	8,988.64
Koramangala	Milk	January 18,2016	1,621.58

LET'S PARTITION IT ON THE PRODUCT COLUMN

THERE ARE 4 DISTINCT
VALUES IN PRODUCT COLUMN

4 PARTITIONS OF THIS TABLE

Product
Bananas
Nutella
Peanut Butter
Milk

LET'S PARTITION IT ON THE PRODUCT COLUMN

StoreLocation	Product	Date	Revenue
Bellandur	Bananas	January 18,2016	8,236.33
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
Koramangala	Nutella	January 18,2016	3,644.33
Koramangala	Peanut Butter	January 18,2016	8,988.64
Koramangala	Milk	January 18,2016	1,621.58

4 PARTITIONS OF THIS TABLE

PRODUCT = BANANAS

StoreLocation	Date	Revenue
Bellandur	January 18,2016	8,236.33
Koramangala	January 18,2016	9,456.01

PRODUCT = PEANUT BUTTER

StoreLocation	Date	Revenue
Bellandur	January 18,2016	5,316.89
Koramangala	January 18,2016	8,988.64

PRODUCT = NUTELLA

StoreLocation	Date	Revenue
Bellandur	January 18,2016	7455.67
Koramangala	January 18,2016	3644.33

PRODUCT = MILK

StoreLocation	Date	Revenue
Bellandur	January 18,2016	2,433.76
Koramangala	January 18,2016	1,621.58

4 PARTITIONS OF THIS TABLE

PRODUCT = BANANAS

StoreLocation	Date	Revenue
Bellandur	January 18,2016	8,236.33
Koramangala	January 18,2016	9,456.01

PRODUCT = PEANUT BUTTER

StoreLocation	Date	Revenue
Bellandur	January 18,2016	5,316.89
Koramangala	January 18,2016	8,988.64

PRODUCT = NUTELLA

StoreLocation	Date	Revenue
Bellandur	January 18,2016	7455.67
Koramangala	January 18,2016	3644.33

PRODUCT = MILK

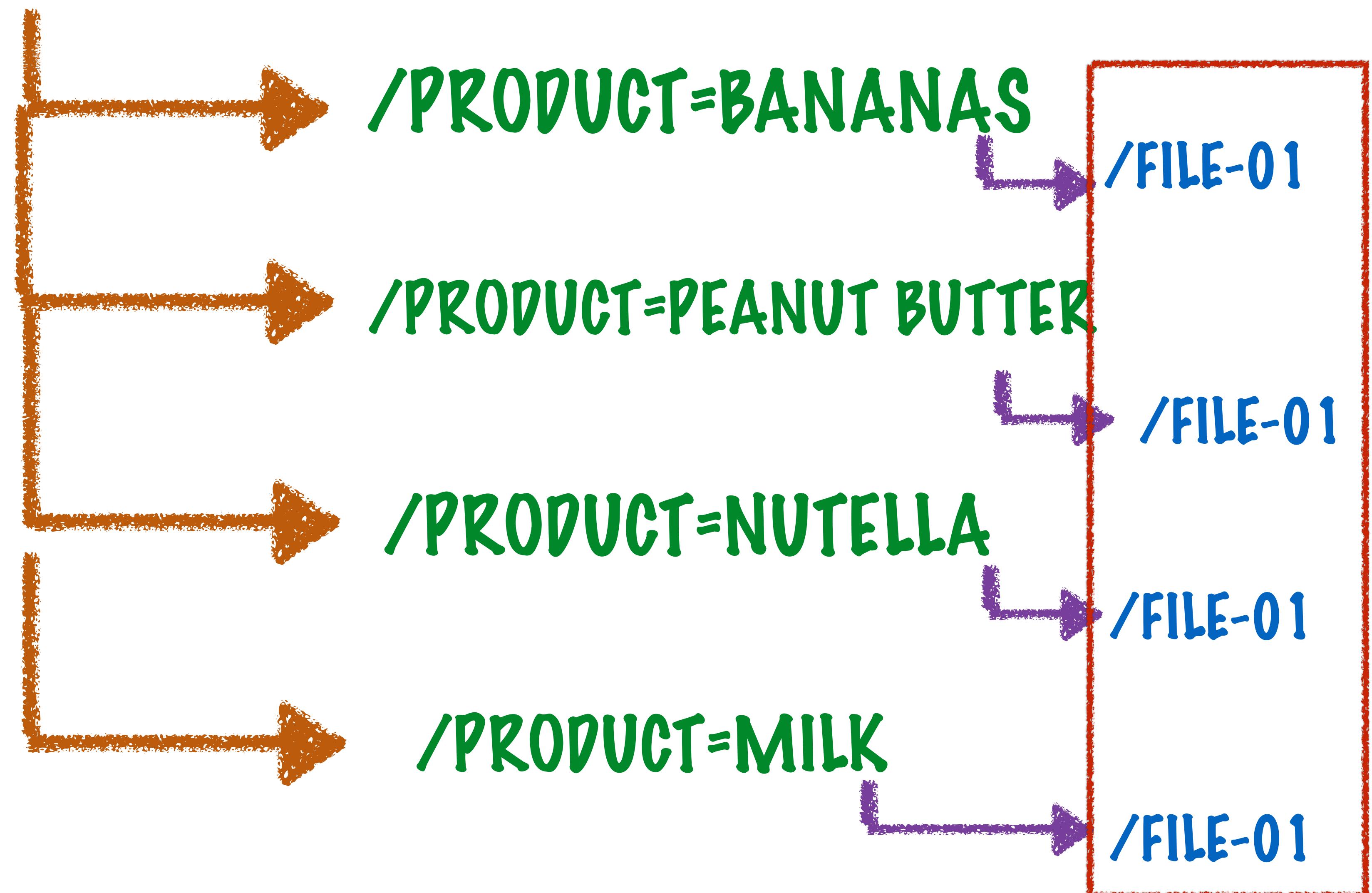
StoreLocation	Date	Revenue
Bellandur	January 18,2016	2,433.76
Koramangala	January 18,2016	1,621.58

**THESE BECOME 4 SUB-DIRECTORIES
IN THE TABLE'S DIRECTORY**

/USER/HIVE/WAREHOUSE

→ /SALES-TABLE

DATA IS STORED IN
THESE FILES



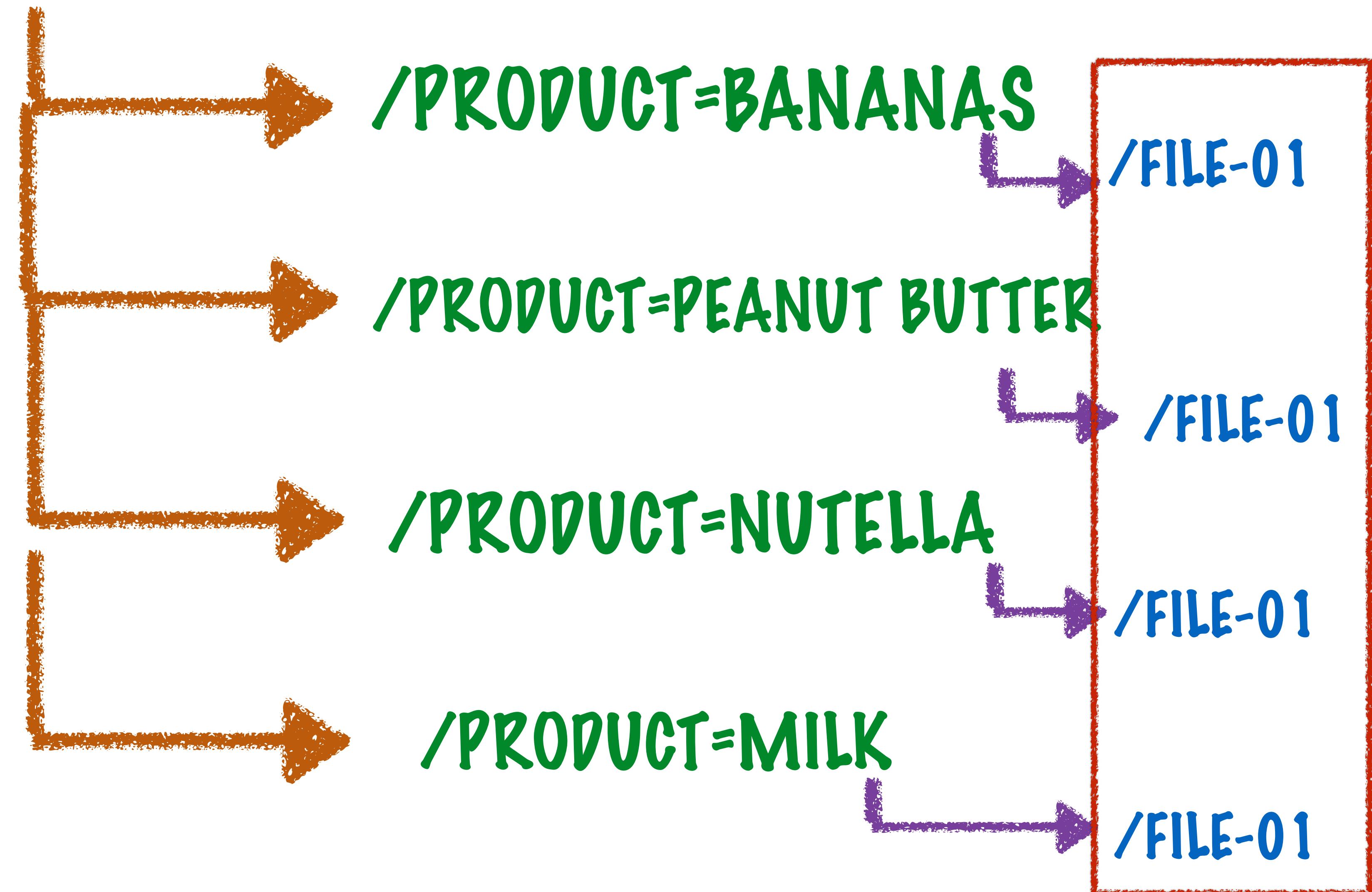
/USER/HIVE/WAREHOUSE



/SALES-TABLE

THERE IS JUST ONE FILE
INSIDE EACH DIRECTORY

BUT THAT
NEED NOT BE
THE CASE -
YOU CAN HAVE
MULTIPLE
FILES IN A
DIRECTORY AS
WELL



WHY SHOULD WE PARTITION TABLES?

IMPROVED PERFORMANCE

LOGICAL ORGANIZATION OF DATA

PARTITIONING DETERMINES DATA STORAGE
STRUCTURES AND SUBDIRECTORIES

WHY SHOULD WE PARTITION TABLES?

PARTITIONING DETERMINES DATA STORAGE
STRUCTURES AND SUBDIRECTORIES

PARTITIONING IMPROVES QUERY
PERFORMANCE

WHY SHOULD WE PARTITION TABLES? PARTITIONING IMPROVES QUERY PERFORMANCE

StoreLocation	Product	Date	Revenue
Bellandur	Bananas	January 18,2016	8,236.33
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
Koramangala	Nutella	January 18,2016	3,644.33
Koramangala	Peanut Butter	January 18,2016	8,988.64
Koramangala	Milk	January 17,2016	1,621.58

CALCULATE TOTAL REVENUE FROM SELLING
MILK ON JANUARY 17

IN AN UNPARTITIONED TABLE, WE
HAVE TO SCAN THE ENTIRE TABLE

PARTITIONING IMPROVES QUERY PERFORMANCE

CALCULATE TOTAL REVENUE FROM SELLING MILK ON JANUARY 17

THIS IS HOW OUR PARTITIONED TABLE LOOKS

PRODUCT = BANANAS

StoreLocation	Date	Revenue
Bellandur	January 18,2016	8,236.33
Koramangala	January 18,2016	9,456.01

PRODUCT = PEANUT BUTTER

StoreLocation	Date	Revenue
Bellandur	January 18,2016	5,316.89
Koramangala	January 18,2016	8,988.64

PRODUCT = NUTELLA

StoreLocation	Date	Revenue
Bellandur	January 18,2016	7455.67
Koramangala	January 18,2016	3644.33

PRODUCT = MILK

StoreLocation	Date	Revenue
Bellandur	January 18,2016	2,433.76
Koramangala	January 17,2016	1,621.58

PARTITIONING IMPROVES QUERY PERFORMANCE

CALCULATE TOTAL REVENUE FROM SELLING MILK ON JANUARY 17

THIS IS HOW OUR PARTITIONED TABLE LOOKS

PRODUCT = MILK

StoreLocation	Date	Revenue
Bellandur	January 18,2016	2,433.76
Koramangala	January 17,2016	1,621.58

IN A PARTITIONED TABLE, WE JUST SCAN THE
PARTITION CORRESPONDING TO PRODUCT = MILK

PARTITIONING IMPROVES QUERY PERFORMANCE

CALCULATE TOTAL REVENUE FROM SELLING MILK ON JANUARY 17

THIS IS HOW OUR PARTITIONED TABLE LOOKS

PRODUCT = MILK		
StoreLocation	Date	Revenue
Bellandur	January 18,2016	2,433.76
Koramangala	January 17,2016	1,621.58

THE DATA TO PROCESS IS ROUGHLY 1/4TH OF
THE TOTAL DATA PRESENT IN THIS TABLE

A POTENTIALLY HUGE SAVING!

PARTITIONING IMPROVES QUERY PERFORMANCE

THE PERFORMANCE IMPROVEMENT
CAN BE DRAMATIC

ONLY IF THE PARTITIONING SCHEME REFLECTS
COMMON FILTERING AND COMMON QUERIES

PARTITIONING IMPROVES QUERY PERFORMANCE

IN THE PREVIOUS EXAMPLE, IF WE HAD PARTITIONS BASED ON STORAGE LOCATION, THERE WOULD HAVE BEEN NO ADVANTAGE OF USING PARTITIONS

PARTITIONING IMPROVES QUERY PERFORMANCE

IN THE PREVIOUS EXAMPLE, IF WE HAD PARTITIONS BASED ON STORAGE LOCATION, THERE WOULD HAVE BEEN NO ADVANTAGE OF USING PARTITIONS

ALL PARTITIONS WOULD HAVE TO BE SCANNED TO SEE WHERE MILK WAS SOLD ON JANUARY 17

PARTITIONING IMPROVES QUERY PERFORMANCE

WHAT ARE THE MOST COMMON QUERIES YOU PLAN TO RUN? PLAN YOUR PARTITION BASED ON MAKING THEM FASTER

PARTITIONING IMPROVES QUERY
PERFORMANCE

PARTITIONS ARE A TRADE-OFF

TOO MANY PARTITIONS MAY OPTIMIZE
SOME QUERIES, BUT BE DETRIMENTAL
FOR OTHER IMPORTANT QUERIES

PARTITIONING IMPROVES QUERY
PERFORMANCE

CONSIDER AN ORDER TABLE FOR AN
E-COMMERCE SITE

DO NOT CREATE PARTITIONS BASED ON
CUSTOMER ID

→ THERE WILL BE MILLIONS OF PARTITIONS

PARTITIONING IMPROVES QUERY PERFORMANCE

LARGE NUMBER OF PARTITIONS = LARGE NUMBER OF HADOOP DIRECTORIES

THIS IS A HUGE OVERHEAD FOR THE NAME NODE WHICH MAINTAINS FILE METADATA FOR HADOOP

HOW TO PARTITION TABLES?

BY ADDING

```
partitioned by (Partition_Column_Name column_data_type)
```

TO THE CREATE TABLE COMMAND

```
CREATE TABLE Sales_Data  
(  
    StoreLocation VARCHAR(30),  
    OrderDate DATE,  
    Revenue DECIMAL(10,2)  
)
```

PARTITION
COLUMN IS NOT
MENTIONED
HERE

HOW TO PARTITION TABLES?

BY ADDING

```
partitioned by (Partition_Column_Name column_data_type)
```

TO THE CREATE TABLE COMMAND

```
CREATE TABLE Sales_Data  
(  
    StoreLocation VARCHAR(30),  
    OrderDate DATE,  
    Revenue DECIMAL(10,2)  
)
```

WE WANT TO
PARTITION IT BY
PRODUCT

HOW TO PARTITION TABLES?

BY ADDING

```
partitioned by (Partition_Column_Name column_data_type)
```

```
CREATE TABLE Sales_Data_Product_Partition  
(  
    StoreLocation VARCHAR(30),  
    OrderDate DATE,  
    Revenue DECIMAL(10,2)  
)
```

```
partitioned by (product varchar(30));
```

HOW TO PARTITION TABLES?

NOTE THAT THE COLUMN IS NOT SPECIFIED AS A
PART OF THE CREATE TABLE

```
CREATE TABLE Sales_Data_Product_Partition
(
    StoreLocation VARCHAR(30),
    OrderDate DATE,
    Revenue DECIMAL(10,2)
)
```

partitioned by (product varchar(30));

HOW TO PARTITION TABLES?

INSTEAD IT'S ONLY PRESENT IN THE
SEPARATE PARTITION BY COMMAND

```
CREATE TABLE Sales_Data_Product_Partition
(
    StoreLocation VARCHAR(30),
    OrderDate DATE,
    Revenue DECIMAL(10,2)
)
partitioned by (product varchar(30));
```

IF WE WANT TO PARTITION TABLES BY DATE COLUMN

BY ADDING

partitioned by(column name column_data_type)

```
CREATE TABLE Sales_Data_Date_Partition  
(  
    StoreLocation VARCHAR(30),  
    product VarChar(30),  
    Revenue DECIMAL(10,2)  
)
```

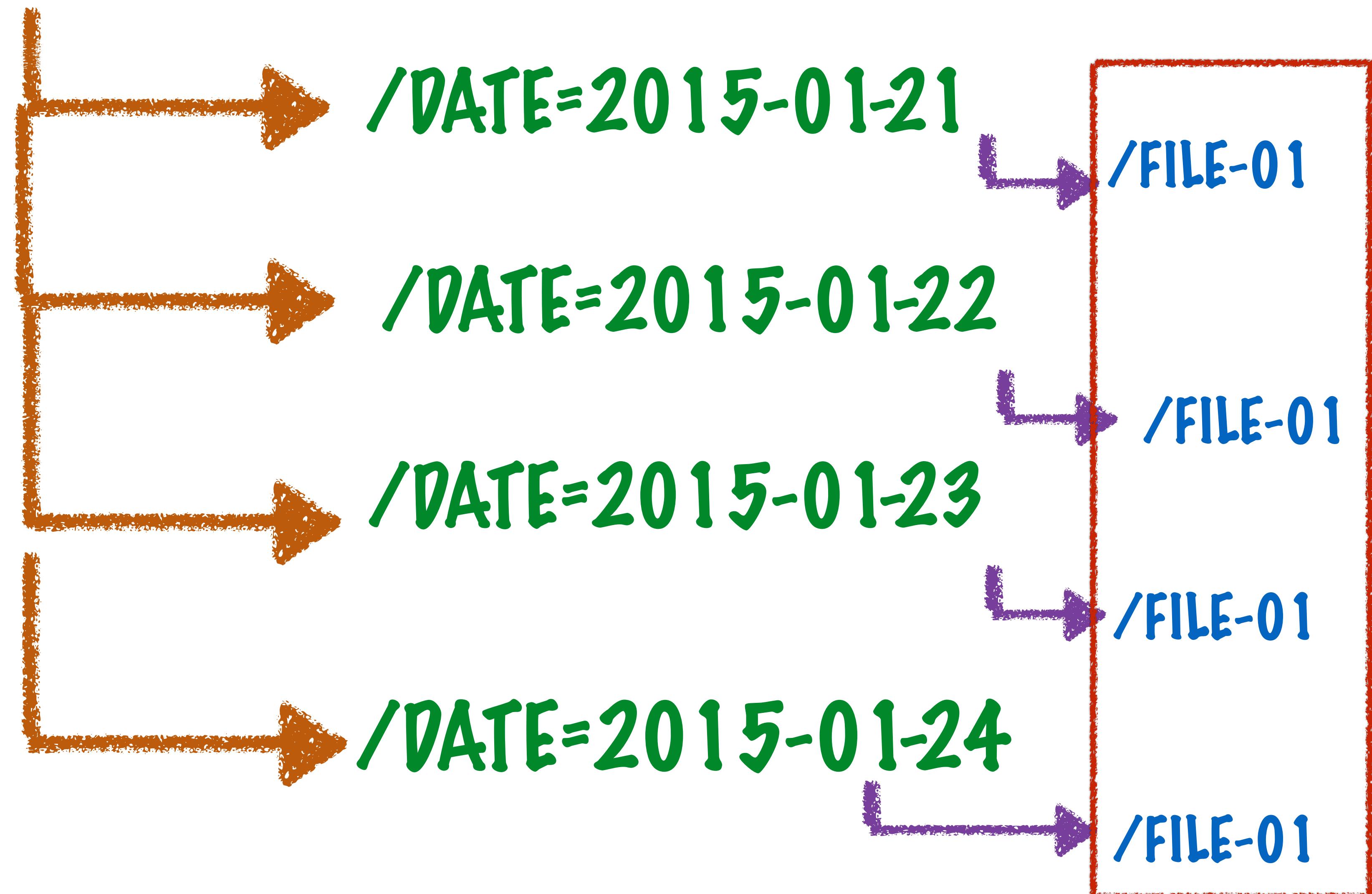
partitioned by(OrderDate DATE);

/USER/HIVE/WAREHOUSE



DATA IS STORED IN THESE FILES

/SALES-DATA-DATE-PARTITION



WE CAN PARTITION TABLES BY TWO COLUMNS

WE CAN PARTITION TABLES BY TWO COLUMNS

```
partitioned by  
(  
    column_name_1 data_type_1,  
    column_name_2 data_type_2  
)
```

WE CAN PARTITION TABLES BY TWO COLUMNS

partitioned by

(

column_name_1 data_type_1 ,

column_name_2 data_type_2

)

THE PARTITIONED BY KEYWORD IS
THE SAME

WE CAN PARTITION TABLES BY TWO COLUMNS

partitioned by

(

column_name_1 data_type_1,

column_name_2 data_type_2

)

EACH COLUMN NAME AND DATA
TYPE IS SPECIFIED

WE CAN PARTITION TABLES BY TWO COLUMNS

partitioned by

(

column_name_1 data_type_1,
column_name_2 data_type_2

)

SEPARATED BY A COMMA

WE CAN PARTITION TABLES BY TWO COLUMNS

```
partitioned by  
(  
    column_name_1 data_type_1,  
    column_name_2 data_type_2  
)
```

THIS COMMAND IS ADDED TO THE CREATE TABLE COMMAND AS WE DID IN LAST SECTION

WE CAN PARTITION TABLES BY TWO COLUMNS

```
partitioned by  
(  
    column_name_1 data_type_1,  
    column_name_2 data_type_2  
)
```

```
CREATE TABLE Sales_Data_Date_Product_Partition  
(  
    StoreLocation VARCHAR(30),  
    Revenue DECIMAL(10,2)  
)
```

partitioned by

```
(  
    OrderDate DATE,  
    product VarChar(30)  
) ;
```

WE CAN PARTITION TABLES BY TWO COLUMNS

```
partitioned by  
(  
    column_name_1 data_type_1,  
    column_name_2 data_type_2  
)  
  
CREATE TABLE Sales_Data_Date_Product_Partition  
(  
    StoreLocation VARCHAR(30),  
    Revenue DECIMAL(10,2)  
)  
  
partitioned by  
(  
    OrderDate DATE,  
    product VarChar(30)  
) ;
```

WE CAN PARTITION TABLES BY TWO COLUMNS

```
partitioned by  
(  
    column_name_1 data_type_1,  
    column_name_2 data_type_2  
)  
  
CREATE TABLE Sales_Data_Date_Product_Partition  
(  
    StoreLocation VARCHAR(30),  
    Revenue DECIMAL(10,2)  
)  
  
partitioned by  
(  
    OrderDate DATE,  
    product VarChar(30)  
) ;
```

WE CAN PARTITION TABLES BY TWO COLUMNS

```
partitioned by  
(  
    column_name_1 data_type_1,  
    column_name_2 data_type_2  
)  
  
CREATE TABLE Sales_Data_Date_Product_Partition  
(  
    StoreLocation VARCHAR(30),  
    Revenue DECIMAL(10,2)  
)  
  
partitioned by  
(  
    OrderDate DATE,  
    product VarChar(30)  
) ;
```

THIS PART IS SAME AS THE CREATE COMMAND

WE CAN PARTITION TABLES BY TWO COLUMNS

LET'S LOOK AT THE DIRECTORY STRUCTURE FOR SUCH A TABLE

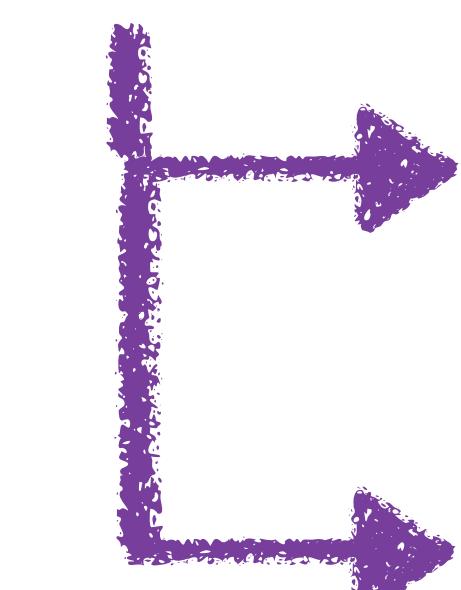
/USER/HIVE/WAREHOUSE



SALES_DATA_DATE_PRODUCT_PARTITION



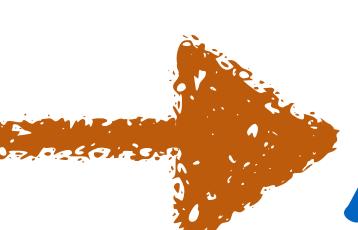
/DATE='2015-01-17'



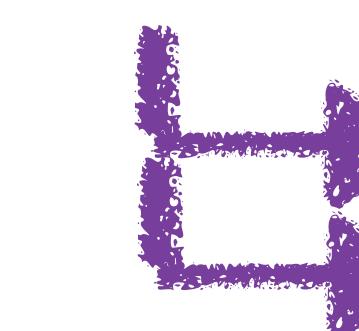
/PRODUCT=BANANAS



/FILE-01



/DATE='2015-01-18'



/PRODUCT = BANANAS

/PRODUCT = PEANUT BUTTER

DATA IS STORED
IN THESE FILES



**HOW DO WE GET DATA FROM
TABLES WITH PARTITIONS?**

HOW DO WE GET DATA FROM TABLES WITH PARTITIONS?

```
CREATE TABLE Sales_Data_Date_Product_Partition  
(  
    StoreLocation VARCHAR(30) ,  
    Revenue DECIMAL(10,2)  
)  
  
partitioned by  
(  
    OrderDate DATE ,  
    product VarChar(30)  
) ;
```

WHEN YOU QUERY
THIS TABLE JUST
TREAT IT AS IF IT
HAS 4 COLUMNS

HOW DO WE GET DATA FROM TABLES WITH PARTITIONS?

partitioned by

(

```
OrderDate DATE,  
product VarChar (30)
```

) ;

YOU CAN WRITE
SELECT STATEMENTS
WHICH TREAT THESE
AS YOU WOULD ANY
REGULAR COLUMN

HOW DO WE GET DATA FROM TABLES WITH PARTITIONS?

partitioned by

(

```
OrderDate DATE,  
product VarChar(30)
```

) ;

QUERIES WITH
CONDITIONS ON THE
PARTITION
COLUMNS WILL
RUN FASTER

**HOW DO WE PUT STUFF INTO
TABLES WITH PARTITIONS?**

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS?

WE CREATED A PARTITIONED TABLE USING THE FOLLOWING COMMAND

```
CREATE TABLE Sales_Data_Date_Partition
(
    StoreLocation VARCHAR(30),
    product VarChar(30),
    Revenue DECIMAL(10,2)
)
partitioned by(OrderDate DATE);
```

```
CREATE TABLE Sales_Data_Date_Partition  
(  
StoreLocation VARCHAR(30) ,  
product VarChar(30) ,  
Revenue DECIMAL(10,2)  
) partitioned by(OrderDate DATE) ;
```

TO INSERT DATA WE WILL USE THE FOLLOWING COMMAND

```
Insert into Sales_Data_Date_Partition  
partition (OrderDate ='2016-01-16')  
values  
( 'Bellandur' , 'Nutella' , 7455.67) ,  
( 'Bellandur' , 'Peanut Butter' , 5316.89) ,  
( 'Bellandur' , 'Milk' , 2433.76) ,  
( 'Koramangala' , 'Bananas' , 9456.01) ;
```

```
CREATE TABLE Sales_Data_Date_Partition  
(  
    StoreLocation VARCHAR(30) ,  
    product VarChar(30) ,  
    Revenue DECIMAL(10,2)  
) partitioned by(OrderDate DATE);
```

SIMILAR TO NORMAL INSERT COMMANDS

```
Insert into Sales_Data_Date_Partition  
partition (OrderDate ='2016-01-16')  
values  
( 'Bellandur' , 'Nutella' , 7455.67) ,  
( 'Bellandur' , 'Peanut Butter' , 5316.89) ,  
( 'Bellandur' , 'Milk' , 2433.76) ,  
( 'Koramangala' , 'Bananas' , 9456.01) ;
```

```
CREATE TABLE Sales_Data_Date_Partition  
(  
    StoreLocation VARCHAR(30) ,  
    product VarChar(30) ,  
    Revenue DECIMAL(10,2)  
)partitioned by (OrderDate DATE) ;
```

SPECIFY THE ACTUAL VALUE OF THE PARTITION WHILE LOADING INTO THE TABLES

```
Insert into Sales Data Date Partition
```

```
partition (OrderDate ='2016-01-16')
```

```
Values
```

```
( 'Bellandur' , 'Nutella' , 7455.67) ,  
( 'Bellandur' , 'Peanut Butter' , 5316.89) ,  
( 'Bellandur' , 'Milk' , 2433.76) ,  
( 'Koramangala' , 'Bananas' , 9456.01) ;
```

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS?

SPECIFY THE ACTUAL VALUE OF THE PARTITION WHILE
LOADING INTO THE TABLES

IF WE HAVE LOTS OF PARTITIONS, THIS
CAN BECOME A MAJOR HEADACHE

SOLUTION: DYNAMIC PARTITIONING

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS?

SOLUTION: DYNAMIC PARTITIONING

THIS CREATES PARTITIONS IN A
PARTITIONED TABLE AUTOMATICALLY

WE CAN LOAD THE ENTIRE DATA IN ONE SWOOP
AND LET HIVE CREATE PARTITIONS

WE NEED NOT CREATE PARTITIONS ONE AT A TIME

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS?

SOLUTION: DYNAMIC PARTITIONING

LET HIVE CREATE PARTITIONS

TO CONFIGURE HIVE TO SUPPORT DYNAMIC PARTITION CREATION, ENTER THE FOLLOWING SET COMMANDS

```
SET hive.exec.dynamic.partition = true;
```

```
SET hive.exec.dynamic.partition.mode = nonstrict;
```

THIS COMMAND CHANGES THE SETTING ONLY FOR A SINGLE SESSION

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS?

```
SET hive.exec.dynamic.partition = true;  
SET hive.exec.dynamic.partition.mode = nonstrict;
```

THIS COMMAND CHANGES THE SETTING ONLY FOR A SINGLE SESSION

TO CHANGE THE SETTINGS PERMANENTLY, EDIT THESE
PROPERTIES IN THE HIVE-SITE.XML FILE

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS?

WHILE SETTING THE DYNAMIC PARTITION MODE, WE SHOULD ALSO SET PARTITION MODE TO NONSTRICT

THERE ARE TWO MODES

STRICT MODE

NONSTRICT MODE

`SET hive.exec.dynamic.partition = true;`

`SET hive.exec.dynamic.partition.mode = nonstrict;`

THIS COMMAND CHANGES SETTING ONLY FOR A SINGLE SESSION

WHILE SETTING THE DYNAMIC PARTITION MODE, WE SHOULD ALSO SET PARTITION MODE TO NONSTRICT

THERE ARE TWO MODES

STRICT MODE

THERE SHOULD BE AT LEAST ONE NON-DYNAMIC PARTITION

NONSTRICT MODE

ALL PARTITIONS ARE ALLOWED TO BE DYNAMIC

```
SET hive.exec.dynamic.partition = true;
```

```
SET hive.exec.dynamic.partition.mode = nonstrict;
```

THERE ARE TWO MODES

STRICT MODE

THERE SHOULD BE AT
LEAST ONE NON-
DYNAMIC PARTITION

NONSTRICT MODE
ALL PARTITIONS ARE
ALLOWED TO BE
DYNAMIC

```
partition (ProductID, OrderDate = '2016-01-16')
```

DYNAMIC PARTITION

NON-DYNAMIC PARTITION

DYNAMIC PARTITIONING

```
SET hive.exec.dynamic.partition = true;  
SET hive.exec.dynamic.partition.mode = nonstrict;
```

ONCE BOTH OF THESE SETTINGS ARE IN PLACE, IT'S EASY TO CHANGE OUR QUERY TO DYNAMICALLY LOAD PARTITIONS.

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS USING DYNAMIC PARTITIONING?

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS USING DYNAMIC PARTITIONING?

LET US IMPORT DATA INTO A PARTITIONED TABLE FROM ANOTHER TABLE

StoreLocation	Product	Date	Revenue
Bellandur	Bananas	January 18,2016	8,236.33
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
Koramangala	Nutella	January 18,2016	3,644.33
Koramangala	Peanut Butter	January 18,2016	8,988.64
Koramangala	Milk	January 18,2016	1,621.58
Bellandur	Bananas	January 17,2016	2342.33
Bellandur	Nutella	January 17,2016	6345.10
Bellandur	Peanut Butter	January 17,2016	5673.01
Bellandur	Milk	January 17,2016	4543.98
Koramangala	Bananas	January 17,2016	8902.65
Koramangala	Nutella	January 17,2016	9114.67
Koramangala	Peanut Butter	January 17,2016	5102.05
Koramangala	Milk	January 17,2016	1299.45

SOURCE TABLE =
SALES_DATA_WITHOUT_PARTITION

LET US IMPORT DATA INTO A PARTITIONED TABLE FROM ANOTHER TABLE

THE TABLE WAS CREATED WITH THE FOLLOWING COMMAND

```
CREATE TABLE Sales_Data_Without_Partition
(
StoreLocation VARCHAR(30) ,
Product VARCHAR(30) ,
OrderDate DATE ,
Revenue DECIMAL(10,2)
);
```

SOURCE TABLE =
SALES_DATA_WITHOUT_PARTITION

StoreLocation	Product	Date	Revenue
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS USING DYNAMIC PARTITIONING?

LET US IMPORT DATA INTO A PARTITIONED TABLE FROM ANOTHER TABLE

StoreLocation	Product	Date	Revenue
Bellandur	Bananas	January 18,2016	8,236.33
	Nutella	January 18,2016	7,455.67
	Peanut Butter	January 18,2016	5,316.89
	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
	Nutella	January 18,2016	3,644.33
	Peanut Butter	January 18,2016	8,988.64
	Milk	January 18,2016	1,621.58
Bellandur	Bananas	January 17,2016	2342.33
	Nutella	January 17,2016	6345.10
	Peanut Butter	January 17,2016	5673.01
	Milk	January 17,2016	4543.98
Koramangala	Bananas	January 17,2016	8902.65
	Nutella	January 17,2016	9114.67
	Peanut Butter	January 17,2016	5102.05
	Milk	January 17,2016	1299.45

SOURCE TABLE =
SALES_DATA_WITHOUT_PARTITION

TABLE IS PARTITIONED
ON DATE AND PRODUCT

DESTINATION TABLE =
SALES_DATE_PRODUCT
_PARTITION

SALES_DATE_PRODUCT_PARTITION IS CREATED LIKE THIS

```
CREATE TABLE Sales_Data_Date_Product_Partition
(
    StoreLocation VARCHAR(30),
    Revenue DECIMAL(10,2)
)

partitioned by
(
    OrderDate DATE,
    product VarChar(30)
);
```

PLEASE KEEP THIS
IN MIND!

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS USING DYNAMIC PARTITIONING?

```
Insert into Sales_Data_Date_Product_Partition  
partition (Product,OrderDate)  
select StoreLocation,Revenue,Product,OrderDate  
from Sales_Data_Without_Partition;
```

SOURCE TABLE =
SALES_DATA_WITHOUT_PARTITION TABLE IS PARTITIONED ON DATE AND PRODUCT

StoreLocation	Product	Date	Revenue
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS USING DYNAMIC PARTITIONING?

SIMILAR TO NORMAL INSERT COMMANDS

```
Insert into Sales_Data_Date_Product_Partition  
partition (Product, OrderDate)  
select StoreLocation, Revenue, Product, OrderDate  
from Sales_Data_Without_Partition;
```

SOURCE TABLE =
SALES_DATA_WITHOUT_PARTITION

TABLE IS PARTITIONED ON DATE AND PRODUCT

StoreLocation	Product	Date	Revenue
Bellandur	Nutella	January 18,2016	7,455.67

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS USING DYNAMIC PARTITIONING?

WHEN WE LOAD INTO PARTITION TABLES, WE
SPECIFY ONLY NAMES OF PARTITION COLUMNS

```
Insert into Sales_Data_Date_Product_Partition  
partition (Product,OrderDate)  
select StoreLocation,Revenue,Product,OrderDate  
from Sales_Data_Without_Partition;
```

IT'S NECESSARY TO INCLUDE THE PARTITION
COLUMNS AS THE LAST COLUMNS IN THE QUERY

SOURCE TABLE =
SALES_DATA_WITHOUT_PARTITION

TABLE IS PARTITIONED ON DATE AND PRODUCT

StoreLocation	Product	Date	Revenue
Bellandur	Nutella	January 18 2016	7 455.67

HOW DO WE PUT STUFF INTO TABLES WITH PARTITIONS USING DYNAMIC PARTITIONING?

```
Insert into Sales_Data_Date_Product_Partition  
partition (Product,OrderDate)  
select StoreLocation,Revenue,Product,OrderDate  
from Sales_Data_Without_Partition;
```

**THIS IS THE QUERY FOR DATA
FROM THE SOURCE TABLE**

SOURCE TABLE =
SALES_DATA_WITHOUT_PARTITION

TABLE IS PARTITIONED ON DATE AND PRODUCT

StoreLocation	Product	Date	Revenue
Bellandur	Nutella	January 18 2016	7 455.67

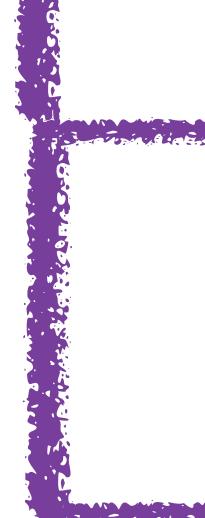
/USER/HIVE/WAREHOUSE



SALES_DATA_DATE_PRODUCT_PARTITION



/DATE='2015-01-17'



/PRODUCT=BANANAS

/FILE-01

/PRODUCT = PEANUT BUTTER

/DATE='2015-01-18'

/PRODUCT = NUTELLA
/PRODUCT = MILK

/DATE='2015-01-19'

/PRODUCT = MILK
/PRODUCT=BANANAS

/DATE='2015-01-20'

/PRODUCT = PEANUT BUTTER

DATA IS STORED
IN THESE FILES



HOW TO CHECK WHAT PARTITIONS EXIST IN A TABLE?

BY USING THE COMMAND

```
show partitions table_name;
```

TO SEE PARTITIONS OF SALES_DATA_PRODUCT_PARTITION

```
hive> SHOW PARTITIONS SALES_DATA_PRODUCT_PARTITION;
OK
product=Bananas
product=Milk
product=Nutella
product=Peanut Butter
```

HOW TO CHECK WHAT PARTITIONS EXIST IN A TABLE?

YOU CAN CHECK THESE PARTITIONS IN THE HIVE WAREHOUSE DIRECTORY

```
hadoop fs -ls /user/hive/warehouse/sales_data_product_partition
```

TO SEE PARTITIONS OF SALES_DATA_PRODUCT_PARTITION

```
Navdeeps-MacBook-Pro:hadoop-2.7.2 navdeepsingh$ hadoop fs -ls /user/hive/warehouse/sales_data_product_partition
16/05/20 18:08:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 4 items
drwxrwxr-x  - navdeepsingh supergroup          0 2016-05-20 18:01 /user/hive/warehouse/sales_data_product_partition/product=Bananas
drwxrwxr-x  - navdeepsingh supergroup          0 2016-05-20 18:01 /user/hive/warehouse/sales_data_product_partition/product=Milk
drwxrwxr-x  - navdeepsingh supergroup          0 2016-05-20 18:01 /user/hive/warehouse/sales_data_product_partition/product=Nutella
drwxrwxr-x  - navdeepsingh supergroup          0 2016-05-20 18:01 /user/hive/warehouse/sales_data_product_partition/product=Peanut Butter
```

HOW TO CHECK WHAT PARTITIONS EXIST IN A TABLE?

TO SEE PARTITIONS OF SALES_DATA_DATE_PRODUCT_PARTITION

```
hadoop fs -ls /user/hive/warehouse/sales_data_dateproduct_partition
```

```
0 2016-05-24 00:51 /user/hive/warehouse/sales_data_date_product_partition/orderdate=2016-01-16
0 2016-05-24 00:52 /user/hive/warehouse/sales_data_date_product_partition/orderdate=2016-01-17
0 2016-05-24 00:52 /user/hive/warehouse/sales_data_date_product_partition/orderdate=2016-01-18
```

FIRST YOU GET THE DIRECTORIES FOR DATE PARTITIONS

HOW TO CHECK WHAT PARTITIONS EXIST IN A TABLE?

TO SEE PARTITIONS OF SALES_DATA_DATE_PRODUCT_PARTITION

```
hadoop fs -ls /user/hive/warehouse/sales_data_dateproduct_partition/  
orderdate=2016-01-16
```

```
24 00:52 /user/hive/warehouse/sales_data_date_product_partition/orderdate=2016-01-16/product=Bananas  
24 00:52 /user/hive/warehouse/sales_data_date_product_partition/orderdate=2016-01-16/product=Milk  
24 00:52 /user/hive/warehouse/sales_data_date_product_partition/orderdate=2016-01-16/product=Nutella  
24 00:52 /user/hive/warehouse/sales_data_date_product_partition/orderdate=2016-01-16/product=Peanut
```

WITHIN EACH DATE PARTITION, THERE IS A PARTITION FOR EACH PRODUCT