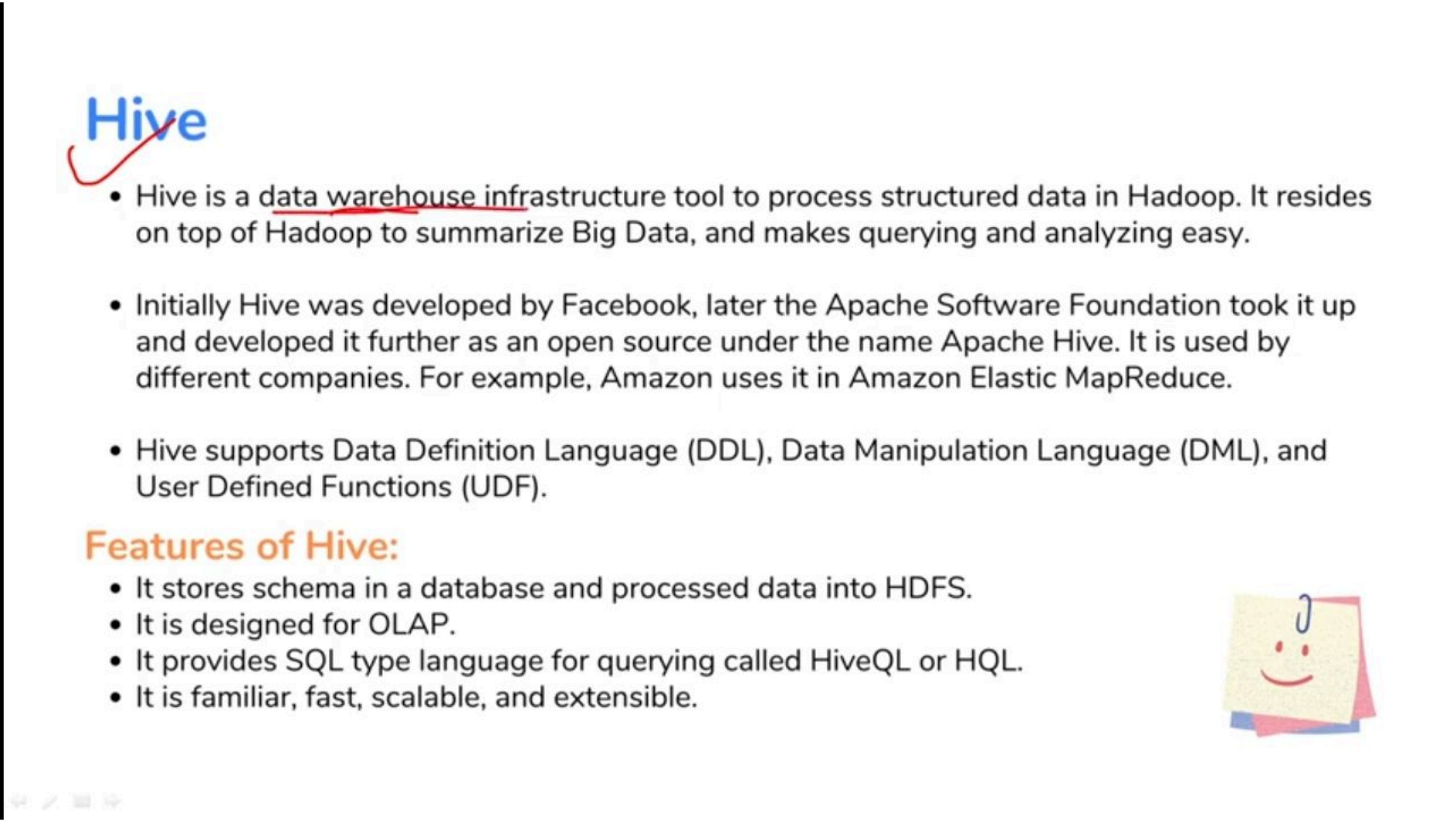


Hive

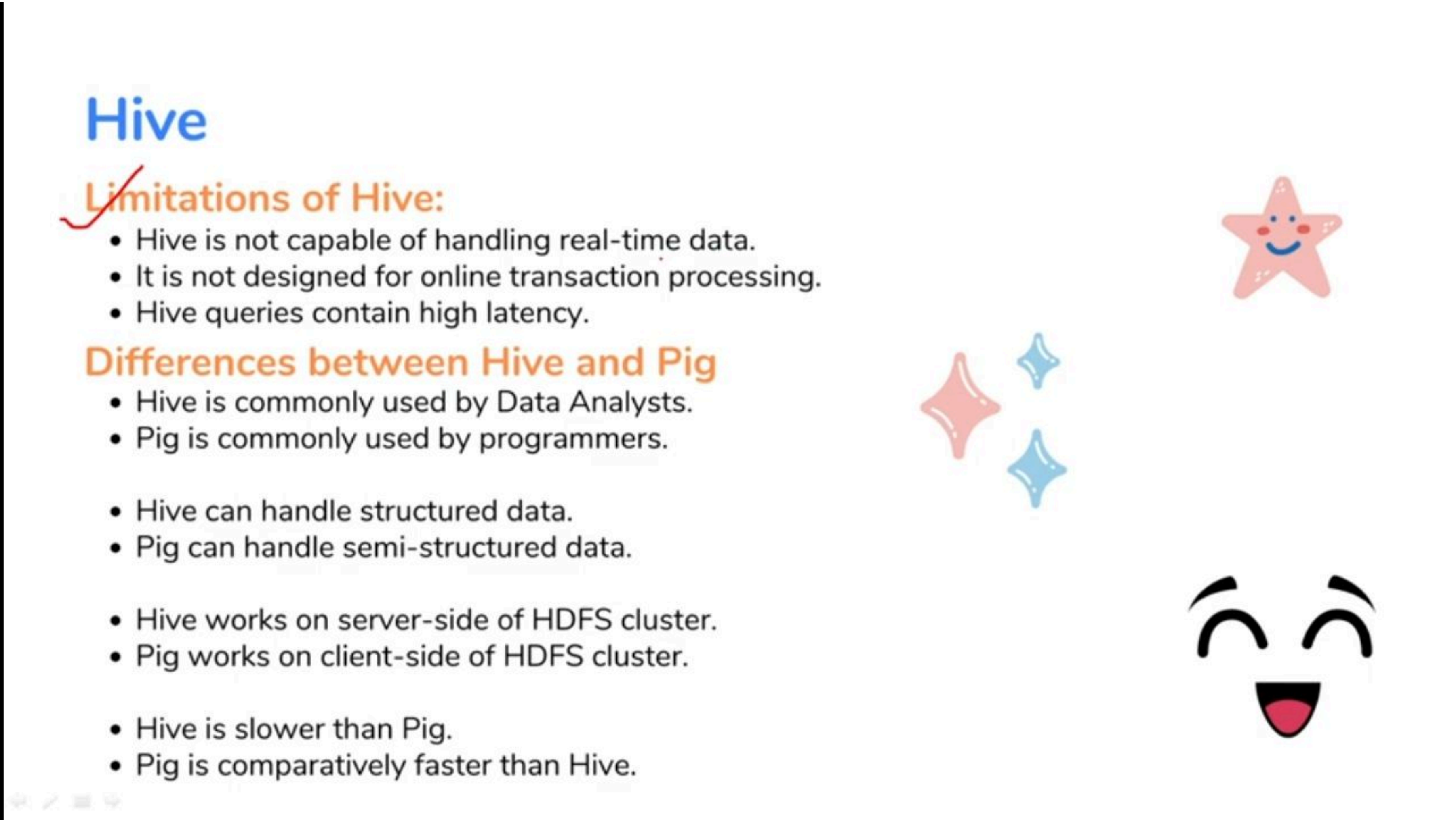
- Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

- Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

- Hive supports Data Definition Language (DDL), Data Manipulation Language (DML), and User Defined Functions (UDF).

Features of Hive:

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.



Hive

Limitations of Hive:

- Hive is not capable of handling real-time data.
- It is not designed for online transaction processing.
- Hive queries contain high latency.

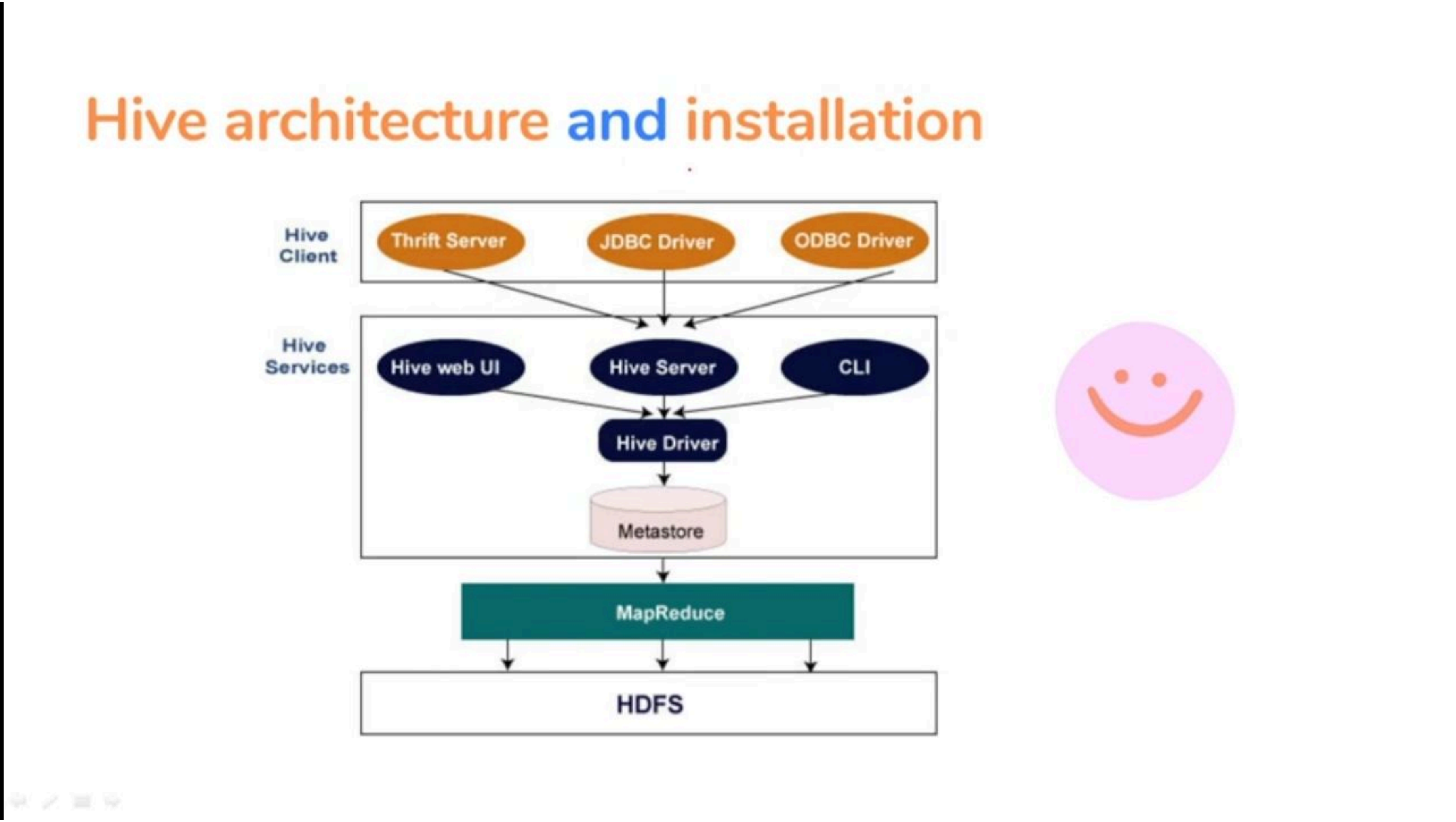
Differences between Hive and Pig

- Hive is commonly used by Data Analysts.
- Pig is commonly used by programmers.

- Hive can handle structured data.
- Pig can handle semi-structured data.

- Hive works on server-side of HDFS cluster.
- Pig works on client-side of HDFS cluster.

- Hive is slower than Pig.
- Pig is comparatively faster than Hive.



Hive architecture and installation

Hive Client

Thrift Server

JDBC Driver

ODBC Driver

Hive Services

Hive web UI

Hive Server

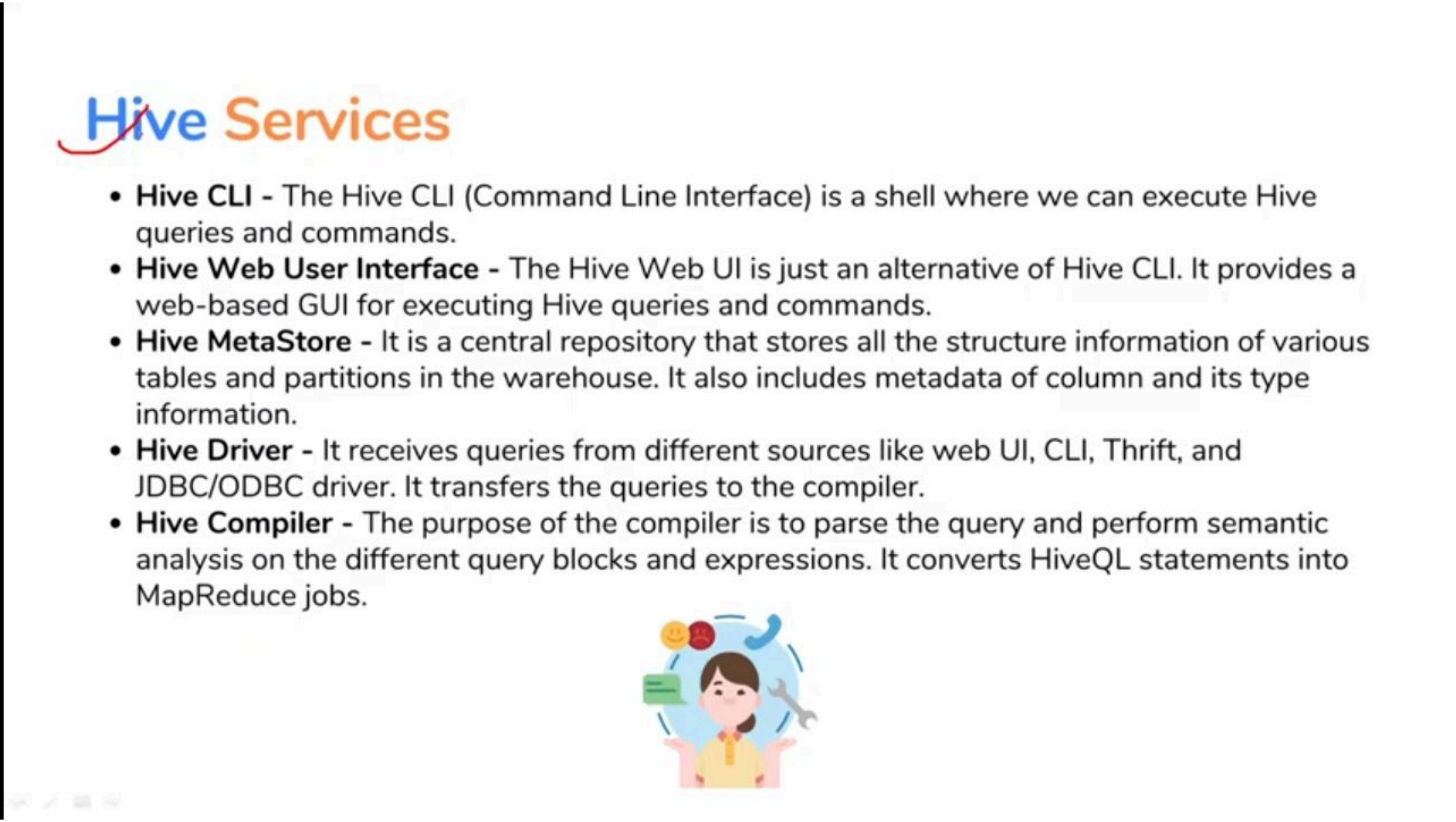
CLI

Hive Driver

Metastore

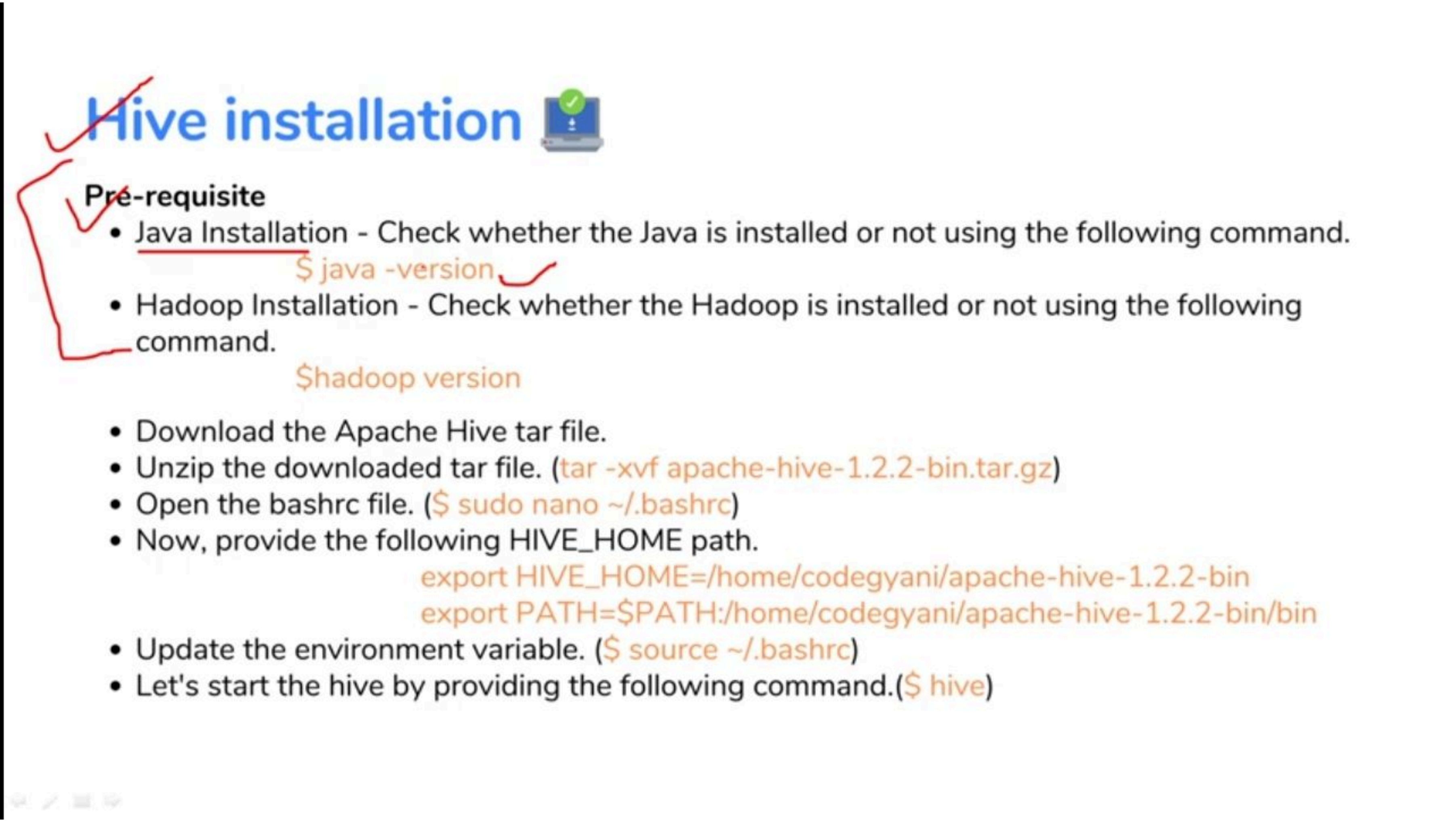
MapReduce

HDFS



Hive Services

- **Hive CLI -** The Hive CLI (Command Line Interface) is a shell where we can execute Hive queries and commands.
- **Hive Web User Interface -** The Hive Web UI is just an alternative of Hive CLI. It provides a web-based GUI for executing Hive queries and commands.
- **Hive MetaStore -** It is a central repository that stores all the structure information of various tables and partitions in the warehouse. It also includes metadata of column and its type information.
- **Hive Driver -** It receives queries from different sources like web UI, CLI, Thrift, and JDBC/ODBC driver. It transfers the queries to the compiler.
- **Hive Compiler -** The purpose of the compiler is to parse the query and perform semantic analysis on the different query blocks and expressions. It converts HiveQL statements into MapReduce jobs.



Hive installation 💻

Pre-requisite

- Java Installation - Check whether the Java is installed or not using the following command.

```
$ java -version
```

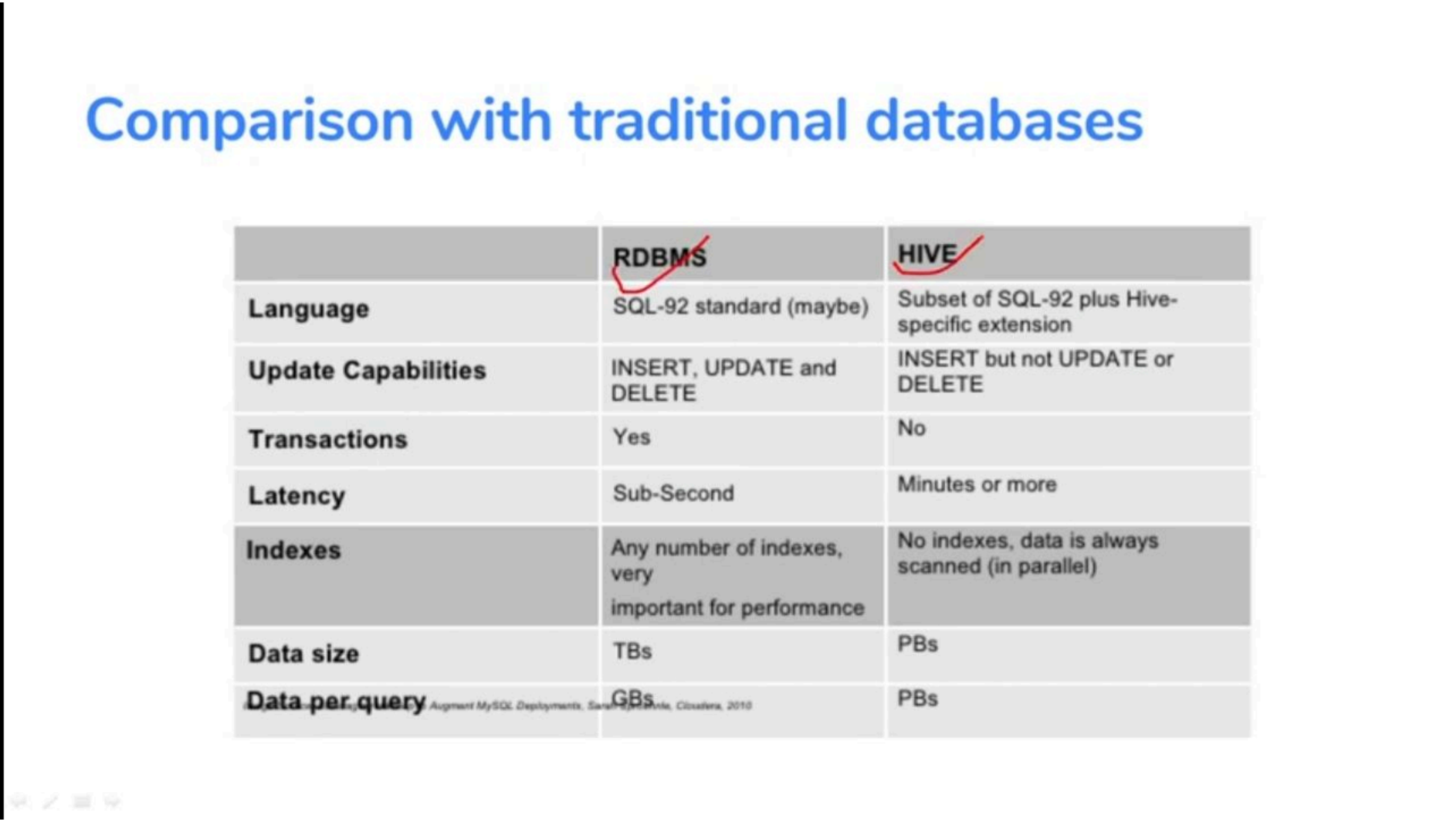
- Hadoop Installation - Check whether the Hadoop is installed or not using the following command.

```
$hadoop version
```

- Download the Apache Hive tar file.
- Unzip the downloaded tar file. (tar -xvf apache-hive-1.2.2-bin.tar.gz)
- Open the bashrc file. ($ sudo nano ~/.bashrc)
- Now, provide the following HIVE_HOME path.

```
export HIVE_HOME=/home/codegyani/apache-hive-1.2.2-bin
export PATH=$PATH:/home/codegyani/apache-hive-1.2.2-bin/bin
```

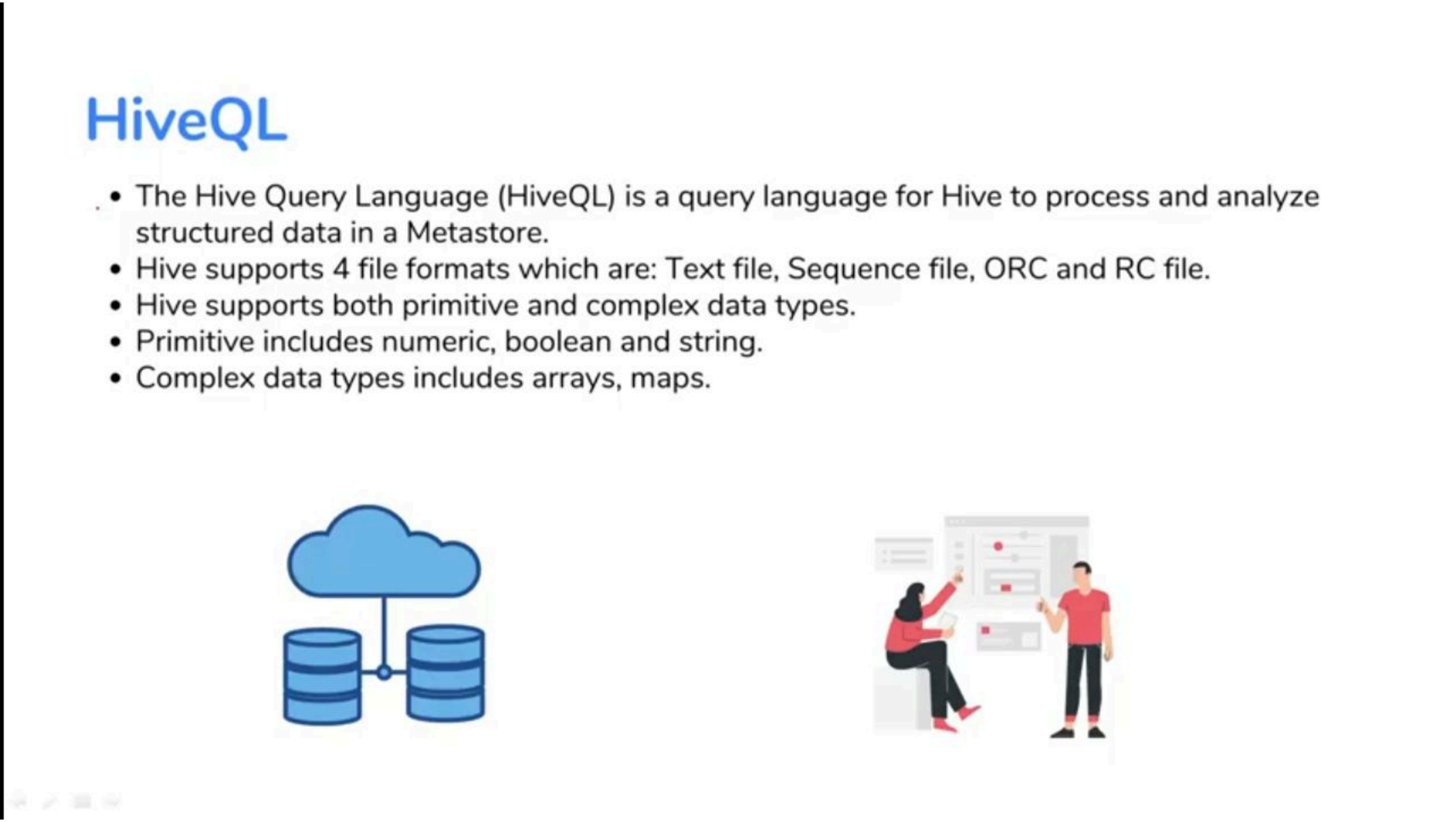
- Update the environment variable. ($ source ~/.bashrc)
- Let's start the hive by providing the following command.($ hive)



Comparison with traditional databases

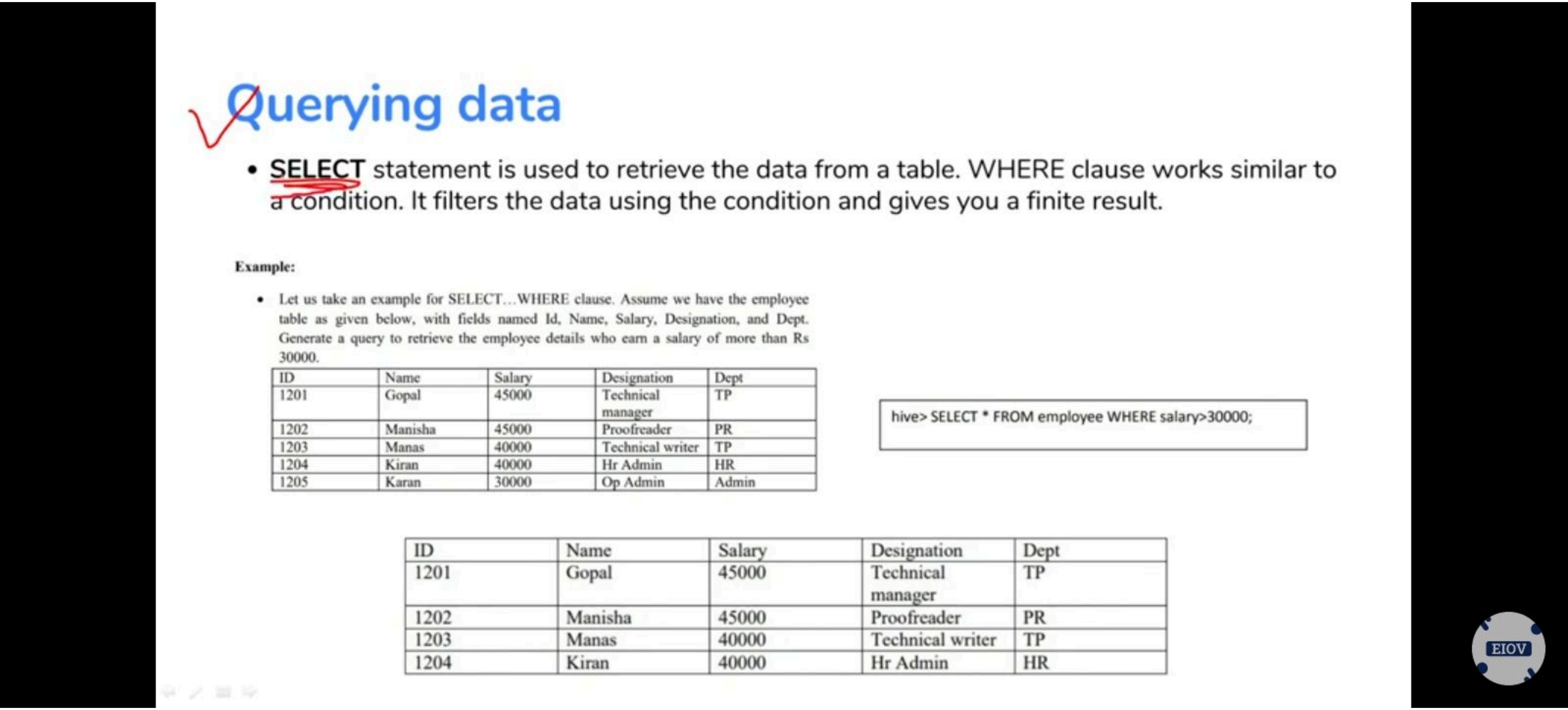
	RDBMS	HIVE
Language	SQL-92 standard (maybe)	Subset of SQL-92 plus Hive-specific extension
Update Capabilities	INSERT, UPDATE and DELETE	INSERT but not UPDATE or DELETE
Transactions	Yes	No
Latency	Sub-Second	Minutes or more
Indexes	Any number of indexes, very important for performance	No indexes, data is always scanned (in parallel)
Data size	TBs	PBs
Data per query	GBs	PBs

[illegible] Augment MySQL Deployments, Sarah Sproehnle, Cloudera, 2010



HiveQL

- The Hive Query Language (HiveQL) is a query language for Hive to process and analyze structured data in a Metastore.
- Hive supports 4 file formats which are: Text file, Sequence file, ORC and RC file.
- Hive supports both primitive and complex data types.
- Primitive includes numeric, boolean and string.
- Complex data types includes arrays, maps.



Querying data

- SELECT statement is used to retrieve the data from a table. WHERE clause works similar to a condition. It filters the data using the condition and gives you a finite result.

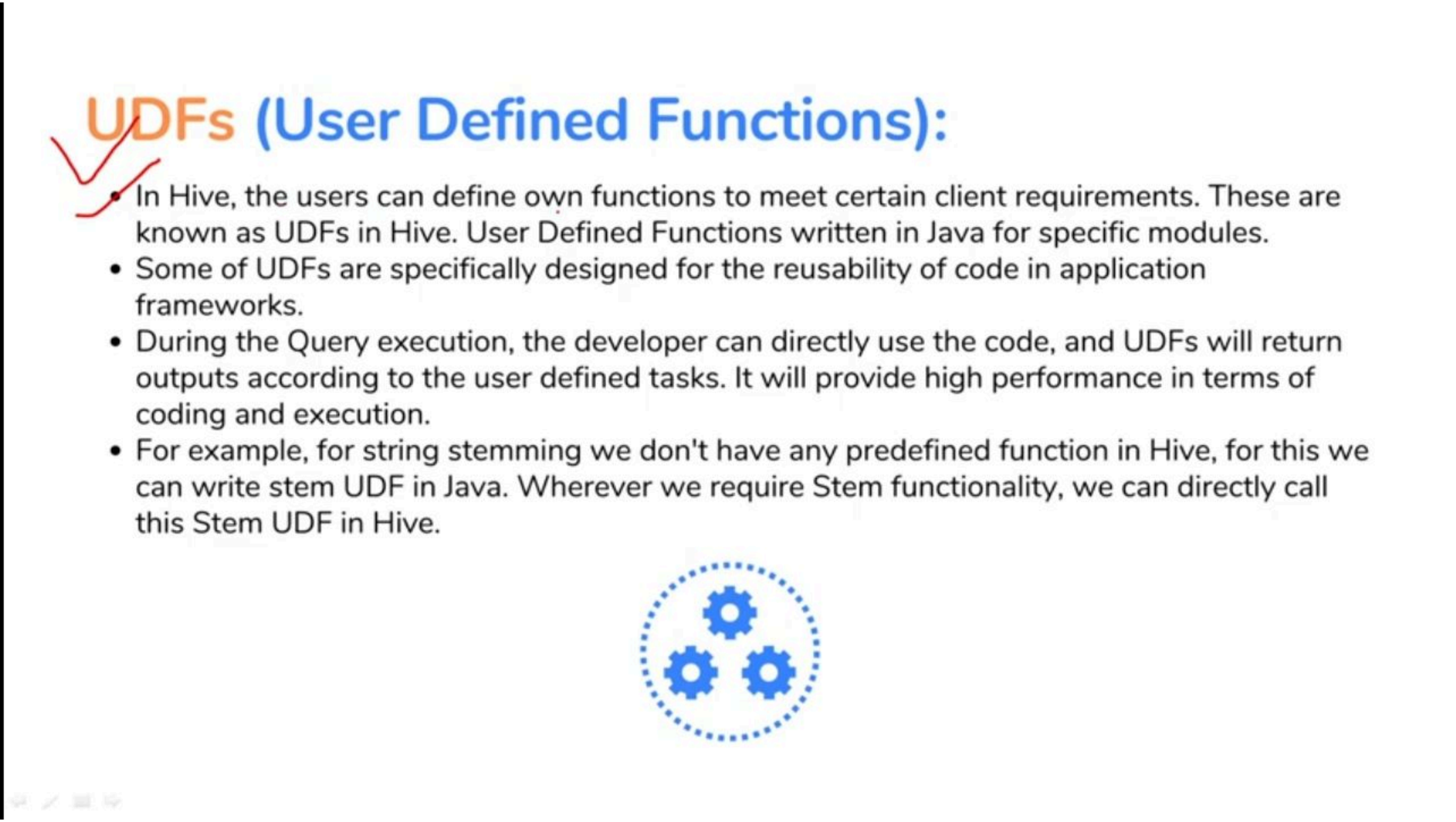
Example:

- Let us take an example for SELECT…WHERE clause. Assume we have the employee table as given below, with fields named Id, Name, Salary, Designation, and Dept. Generate a query to retrieve the employee details who earn a salary of more than Rs 30000.

ID	Name	Salary	Designation	Dept
1201	Gopal	45000	Technical manager	TP
1202	Manisha	45000	Proofreader	PR
1203	Manas	40000	Technical writer	TP
1204	Kiran	40000	Hr Admin	HR
1205	Karan	30000	Op Admin	Admin

```
hive> SELECT * FROM employee WHERE salary>30000;
```

ID	Name	Salary	Designation	Dept
1201	Gopal	45000	Technical manager	TP
1202	Manisha	45000	Proofreader	PR
1203	Manas	40000	Technical writer	TP
1204	Kiran	40000	Hr Admin	HR

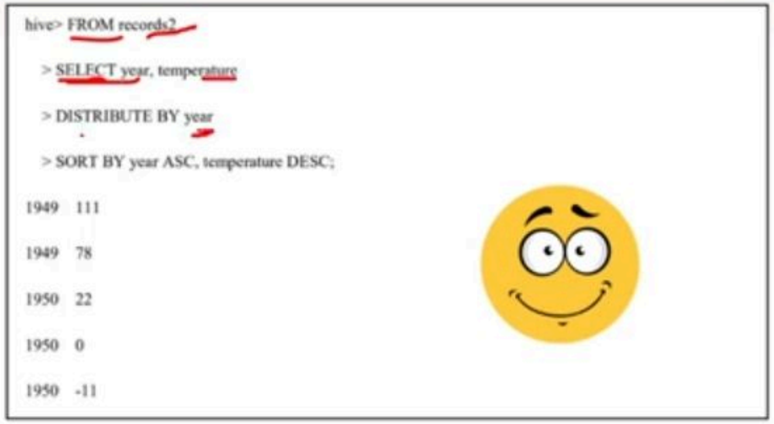


UDFs (User Defined Functions):

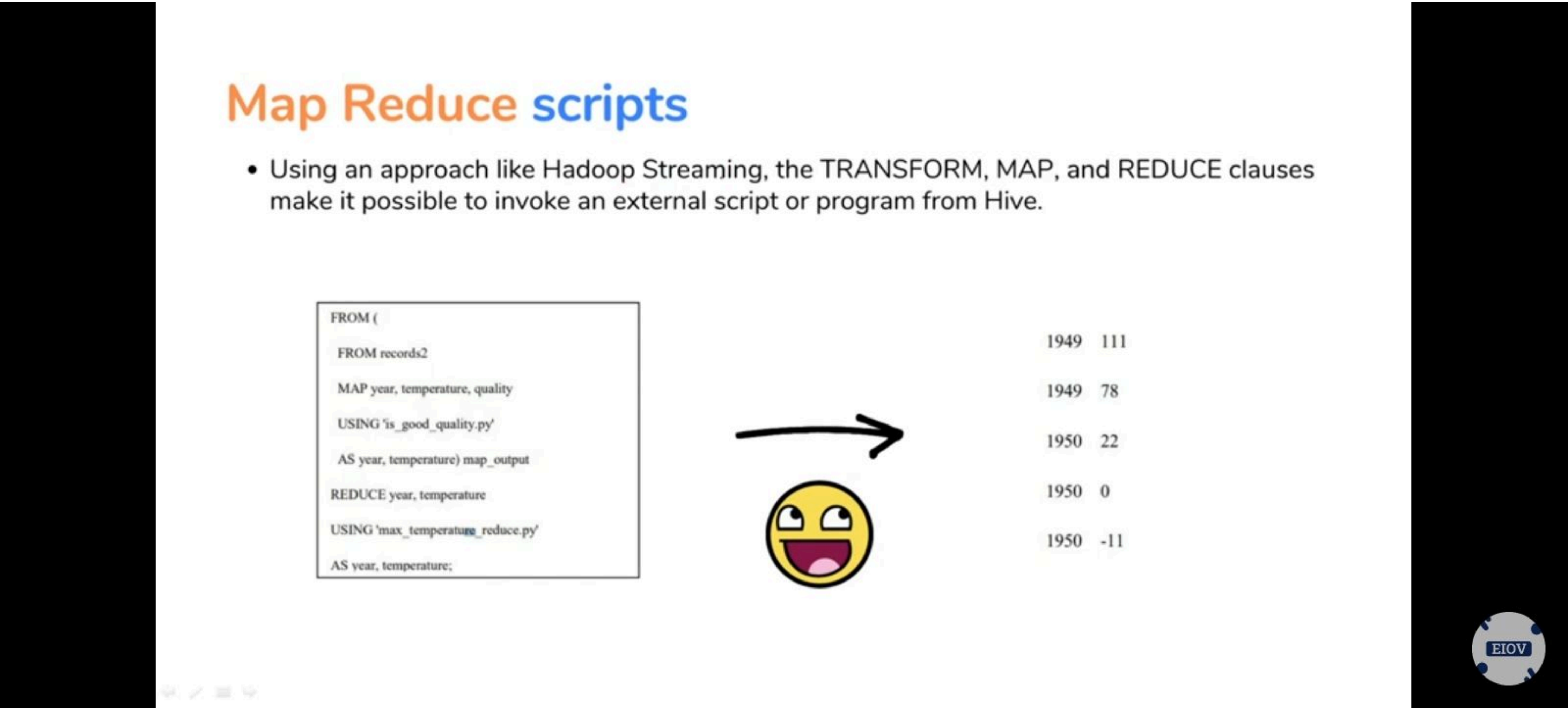
- In Hive, the users can define own functions to meet certain client requirements. These are known as UDFs in Hive. User Defined Functions written in Java for specific modules.
- Some of UDFs are specifically designed for the reusability of code in application frameworks.
- During the Query execution, the developer can directly use the code, and UDFs will return outputs according to the user defined tasks. It will provide high performance in terms of coding and execution.
- For example, for string stemming we don't have any predefined function in Hive, for this we can write stem UDF in Java. Wherever we require Stem functionality, we can directly call this Stem UDF in Hive.

Sorting and aggregating

- Sorting data in Hive can be achieved by use of a standard ORDER BY clause, but there is a catch. ORDER BY produces a result that is totally sorted, as expected, but to do so it sets the number of reducers to one, making it very inefficient for large datasets.
- In some cases, you want to control which reducer a particular row goes to, typically so you can perform some subsequent aggregation. This is what Hive's DISTRIBUTE BY clause does. Here's an example to sort the weather dataset by year and temperature



```
hive> FROM records2
   > SELECT year, temperature
   > DISTRIBUTE BY year
   > SORT BY year ASC, temperature DESC;
1949   111
1949   78
1950   22
1950   0
1950   -11
```

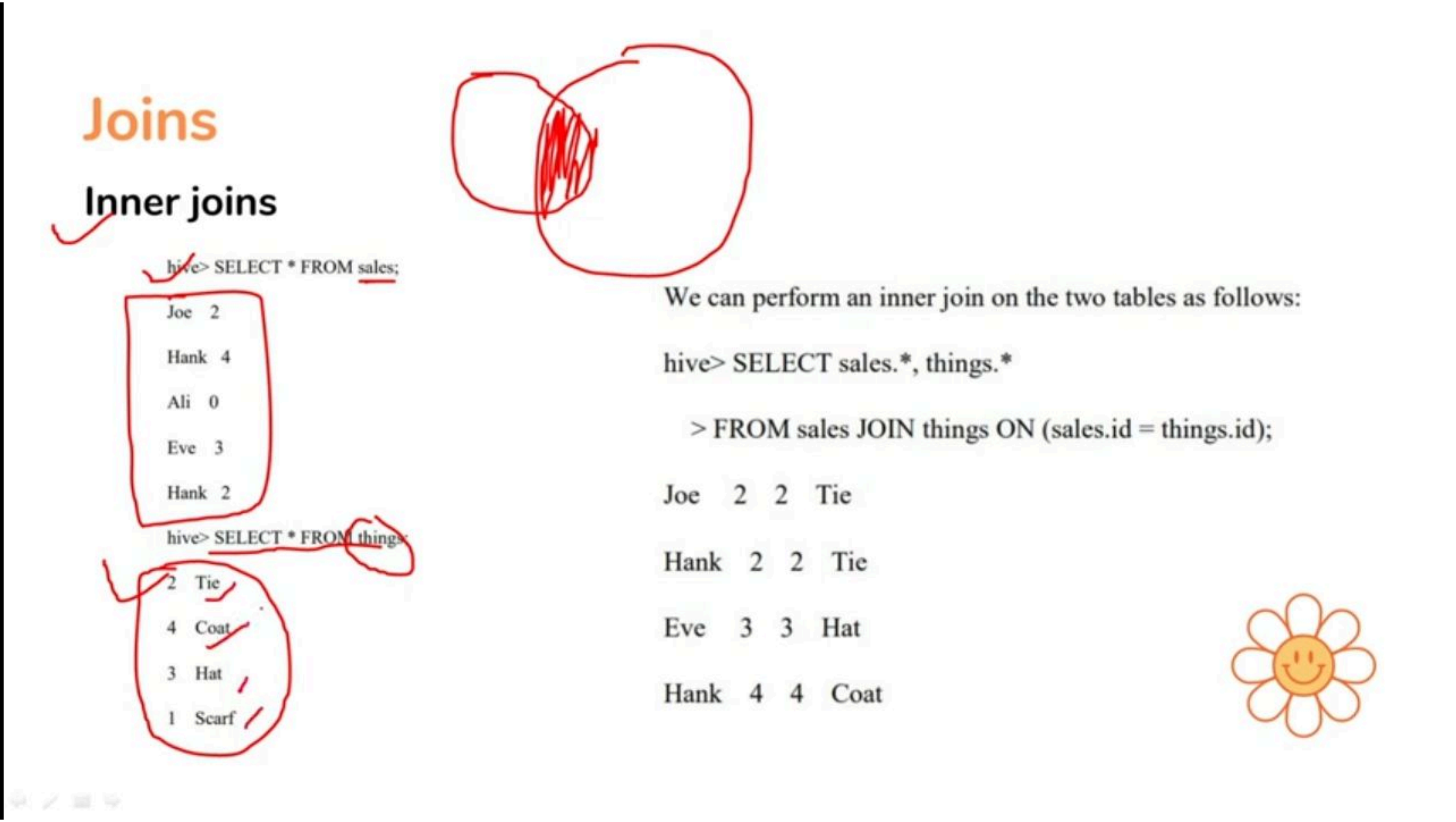


Map Reduce scripts

- Using an approach like Hadoop Streaming, the TRANSFORM, MAP, and REDUCE clauses make it possible to invoke an external script or program from Hive.

```
FROM (
 FROM records2
 MAP year, temperature, quality
 USING 'is_good_quality.py'
 AS year, temperature) map_output
REDUCE year, temperature
USING 'max_temperature_reduce.py'
AS year, temperature;
```

```
1949   111
1949   78
1950   22
1950   0
1950   -11
```



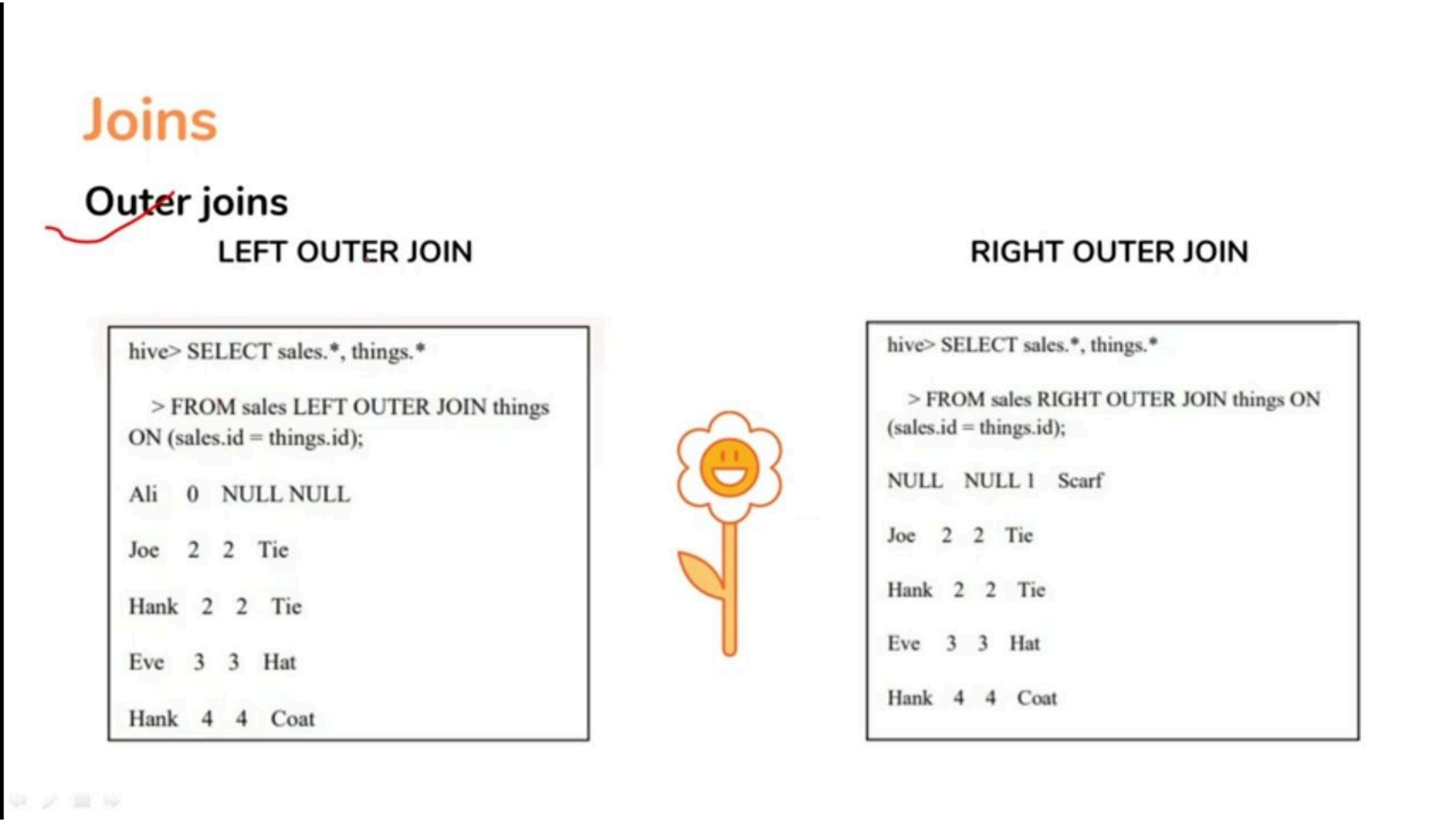
Joins

Inner joins

```
hive> SELECT * FROM sales;
Joe   2
Hank  4
Ali   0
Eve   3
Hank  2
hive> SELECT * FROM things;
2  Tie
4  Coat
3  Hat
1  Scarf
```

We can perform an inner join on the two tables as follows:

```
hive> SELECT sales.*, things.*
   > FROM sales JOIN things ON (sales.id = things.id);
Joe   2  2  Tie
Hank  2  2  Tie
Eve   3  3  Hat
Hank  4  4  Coat
```



Joins

Outer joins

LEFT OUTER JOIN

```
hive> SELECT sales.*, things.*

   > FROM sales LEFT OUTER JOIN things
ON (sales.id = things.id);

Ali    0   NULL NULL

Joe    2   2   Tie

Hank   2   2   Tie

Eve    3   3   Hat

Hank   4   4   Coat
```

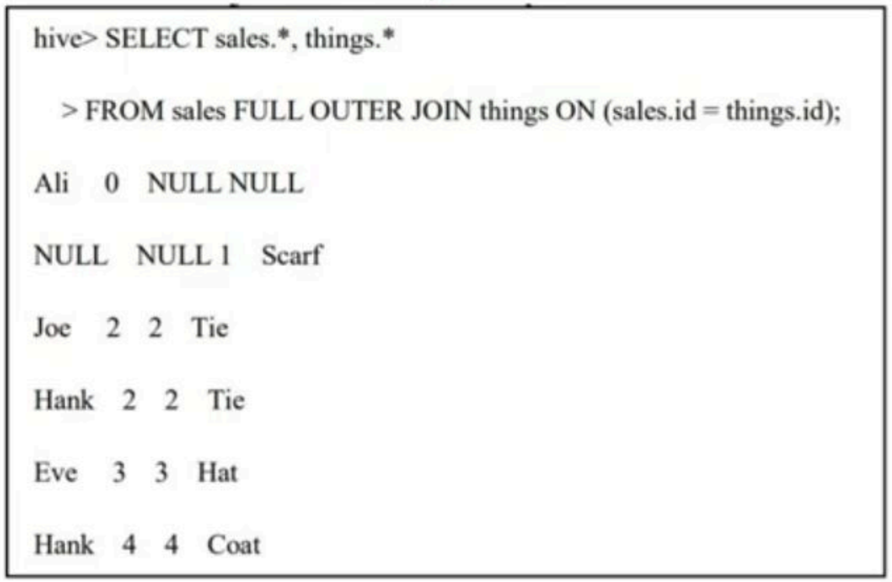
RIGHT OUTER JOIN

```
hive> SELECT sales.*, things.*

   > FROM sales RIGHT OUTER JOIN things ON
(sales.id = things.id);

NULL   NULL 1   Scarf

Joe    2   2   Tie

Hank   2   2   Tie

Eve    3   3   Hat

Hank   4   4   Coat
```

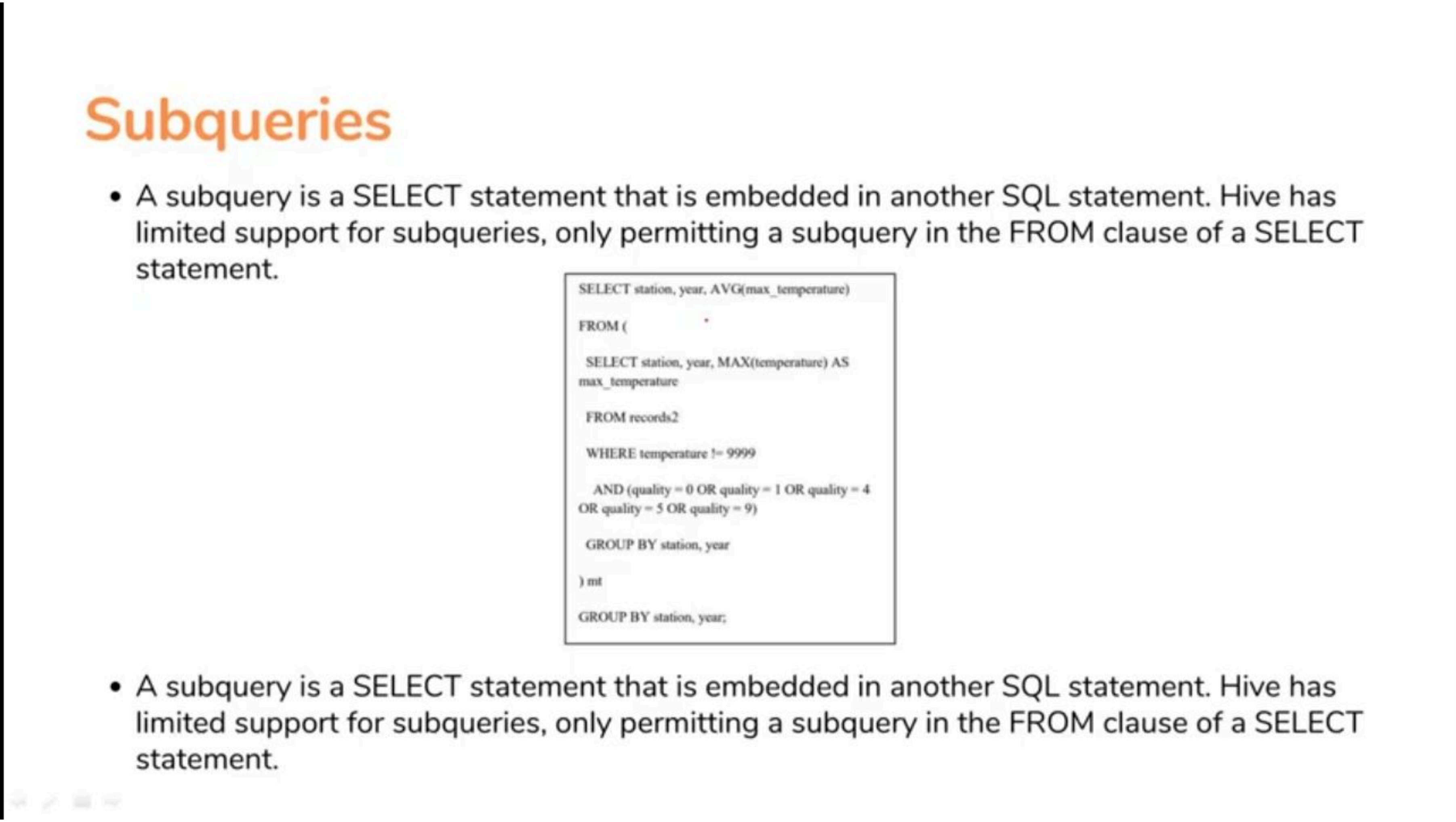
Joins

Outer joins

FULL OUTER JOIN



```
hive> SELECT sales.*, things.*

   > FROM sales FULL OUTER JOIN things ON (sales.id = things.id);

Ali    0    NULL NULL

NULL    NULL 1    Scarf

Joe    2    2    Tie

Hank    2    2    Tie

Eve    3    3    Hat

Hank    4    4    Coat
```



Subqueries

- A subquery is a SELECT statement that is embedded in another SQL statement. Hive has limited support for subqueries, only permitting a subquery in the FROM clause of a SELECT statement.

```
SELECT station, year, AVG(max_temperature)

FROM (

 SELECT station, year, MAX(temperature) AS
max_temperature

 FROM records2

 WHERE temperature != 9999

  AND (quality = 0 OR quality = 1 OR quality = 4
OR quality = 5 OR quality = 9)

 GROUP BY station, year

) mt

GROUP BY station, year;
```

- A subquery is a SELECT statement that is embedded in another SQL statement. Hive has limited support for subqueries, only permitting a subquery in the FROM clause of a SELECT statement.