1. **Develop an EER model for the following:**

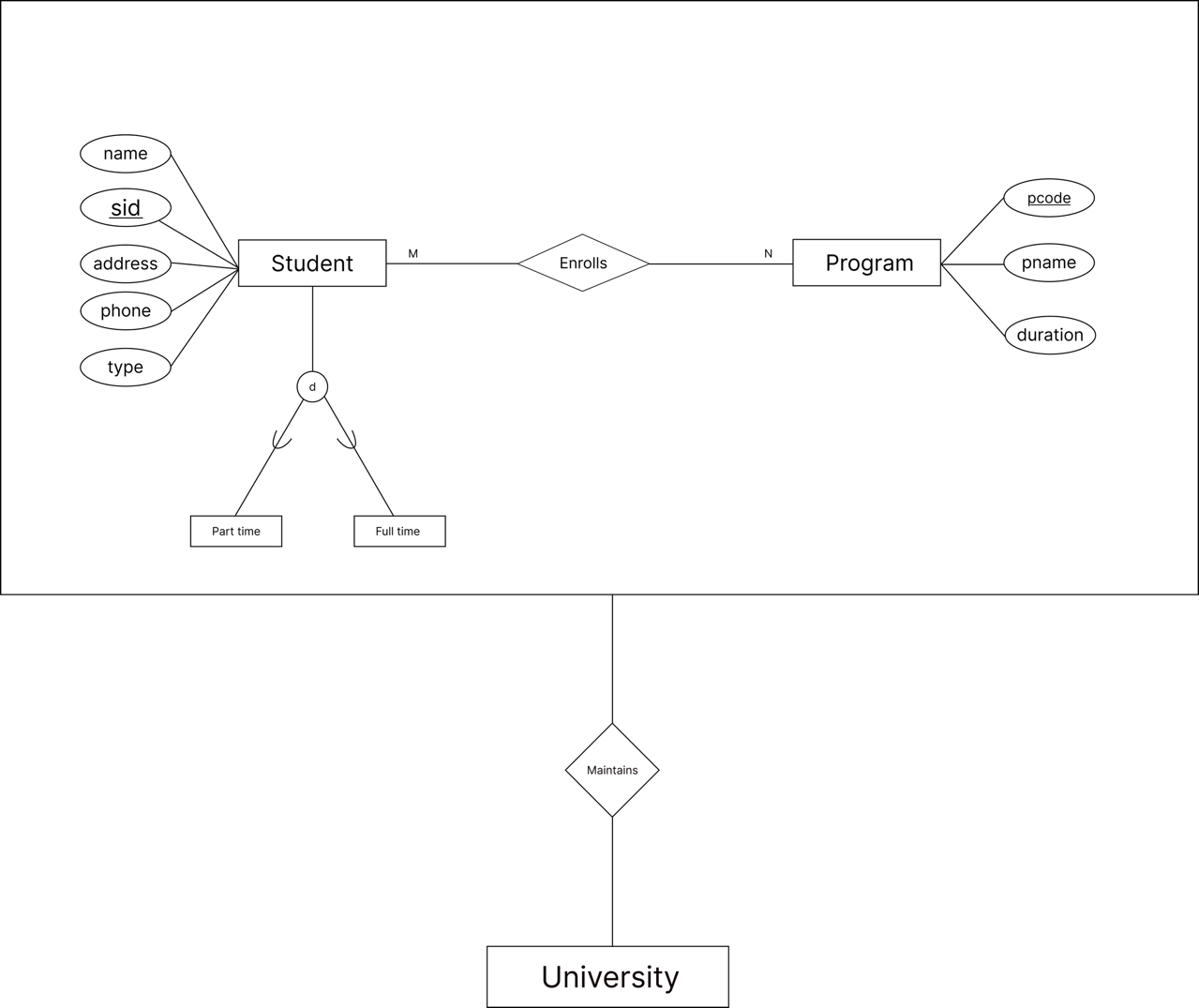
A university maintains records of its students and programs in which they have enrolled.

It stores student id, name, address, and phone number of student and programs code,

program name and duration of a program. A student is either a full-time or a part-time

student (only one of the types). A student can register for many program and programs

can have many students.

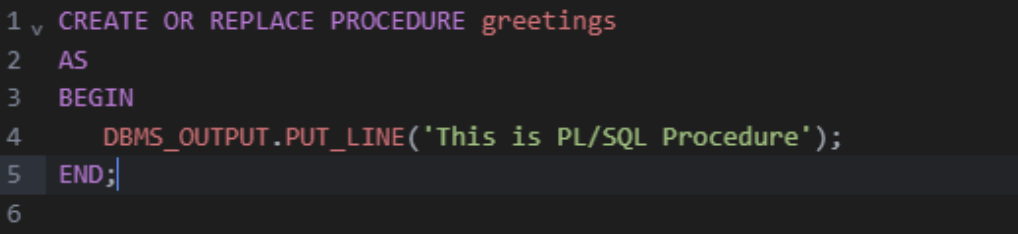


1. **Apply PL/SQL for processing database. Give examples of Procedure, Functions and Triggers.**

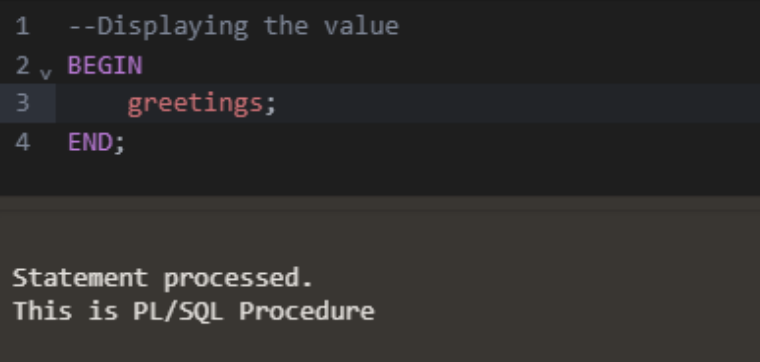
PL/SQL (Procedural Language/Structured Query Language) is a robust extension of SQL used in Oracle databases, enabling procedural programming. With PL/SQL, we can create stored procedures, functions, and triggers to process data within the database efficiently.

Examples:

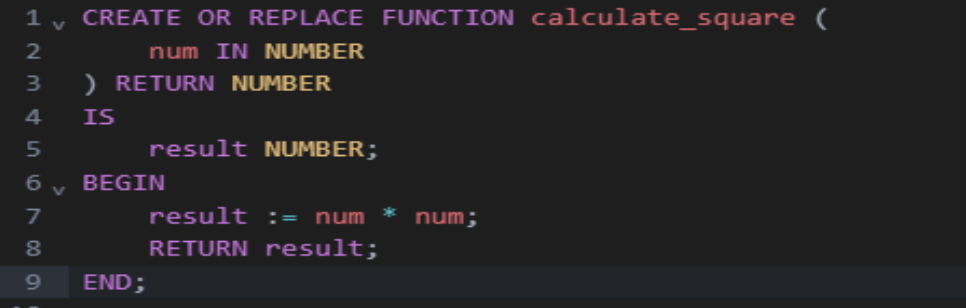
* **Procedure**

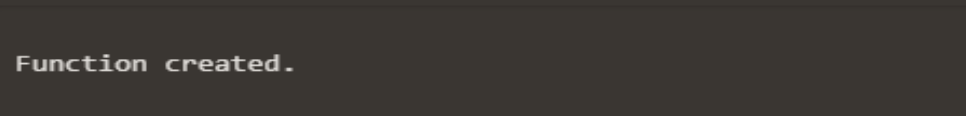
****

**Output:**

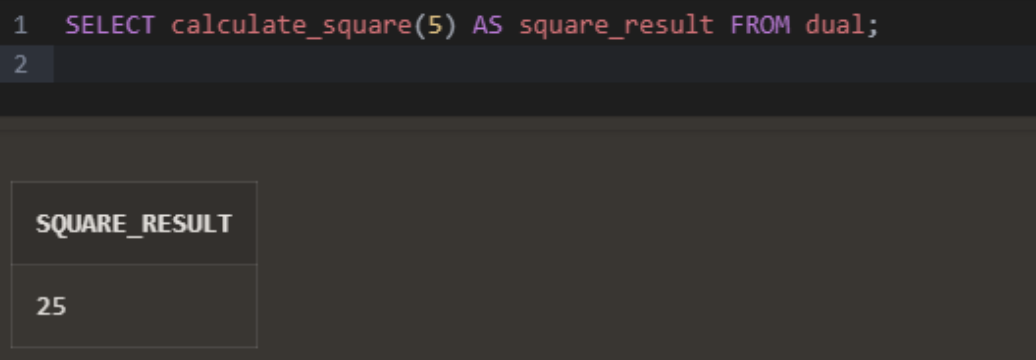


* **Function**

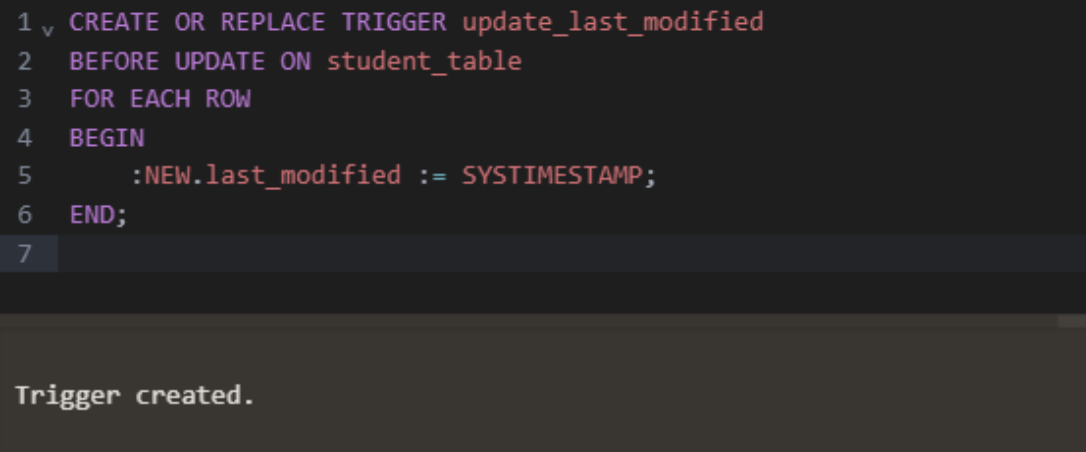
****

****

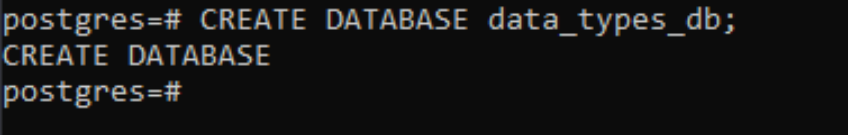
**Output:**

****

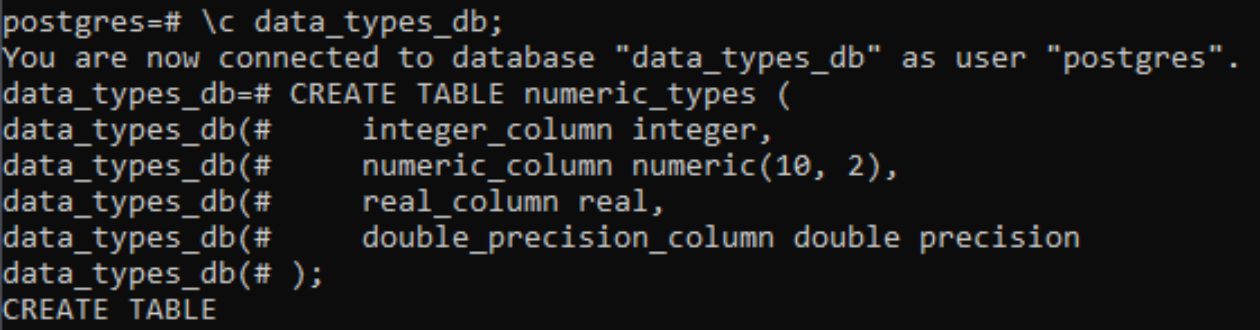
* **Triggers**

****

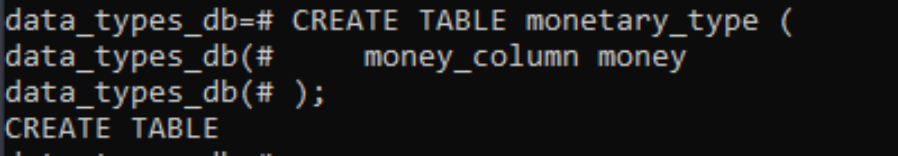
1. **Illustrate different data types in PostgreSQL.**

****

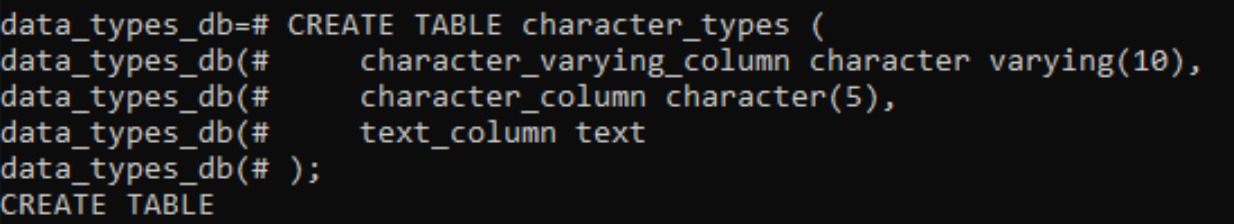
* **Numeric Types**

****

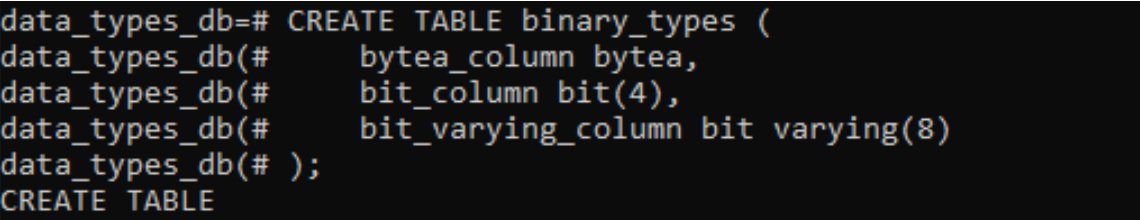
* **Monetary Type**

****

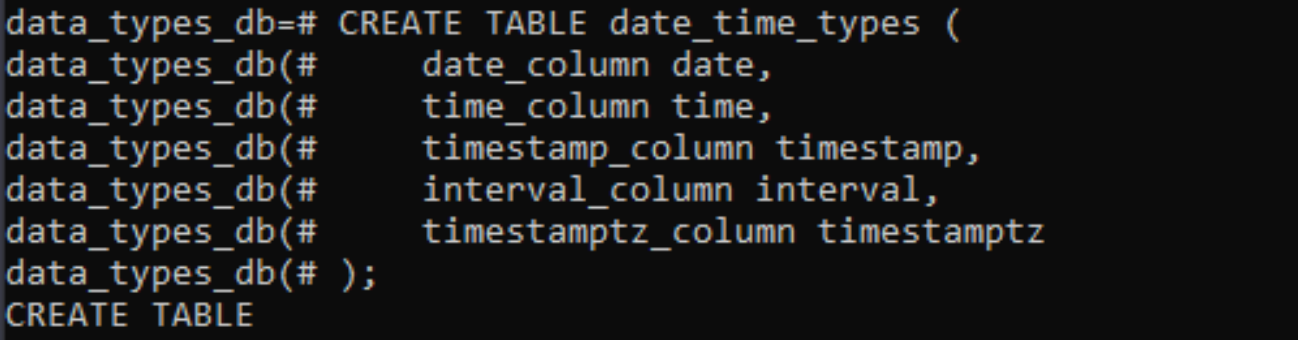
* **Character Type**

****

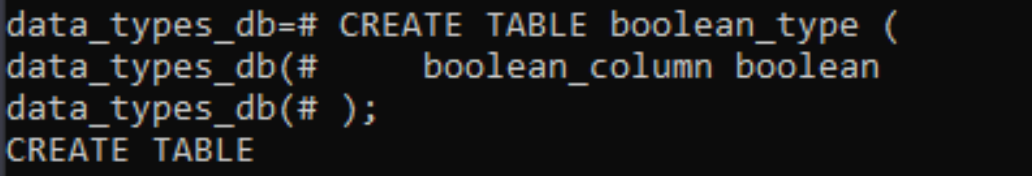
* **Binary Type**

****

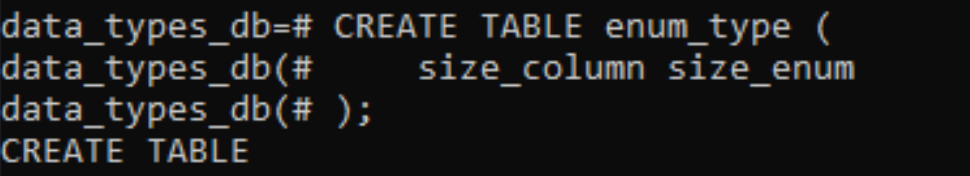
* **Date and Time Type**

****

* **Boolean Type**

****

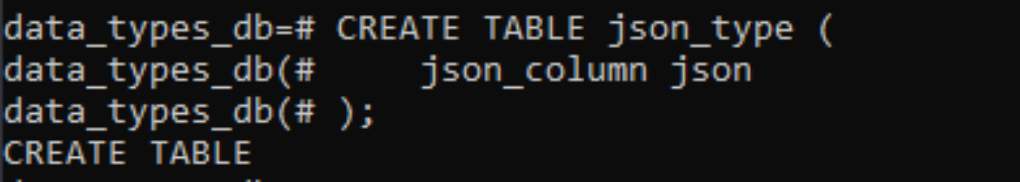
* **Enum Type**

****

* **Network Address Type**

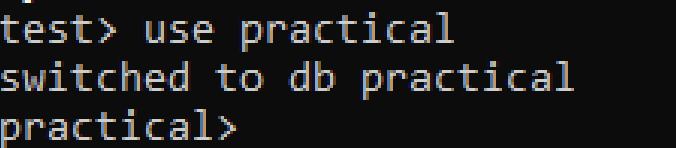
****

* **JSON Type**

****

1. **Apply CRUD operations and retrieve data in NoSQL environment (Use MongoDB or any NoSQL database.**

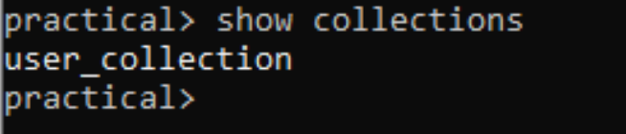
* **Creating a Database**

****

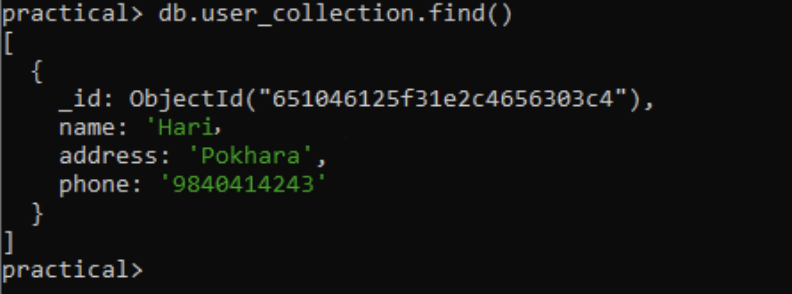
* **Creating a collection and adding a document for a user**

****

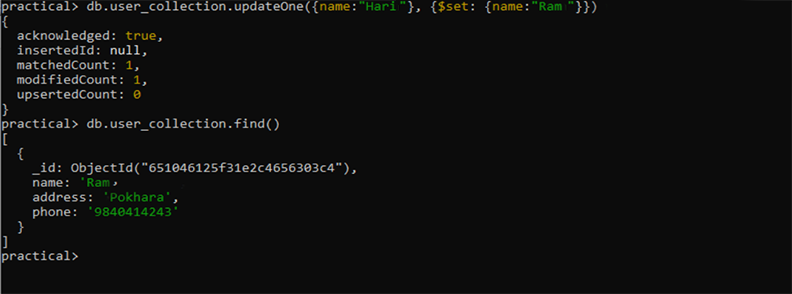
* **Showing the current collection**

****

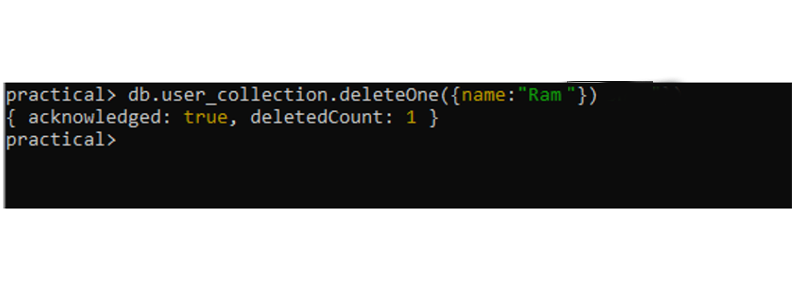
* **Reading the current collection data**

****

* **Updating data**

****

* **Deleting data**

****

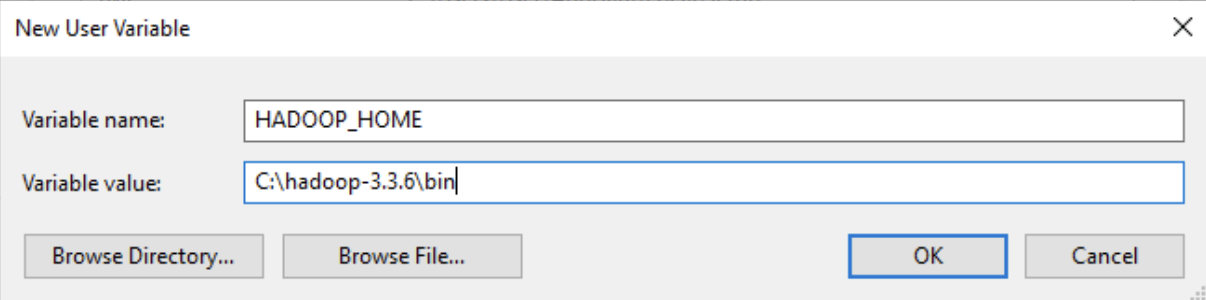
1. **Understand the basic storage architecture of distributed file systems. Setup**

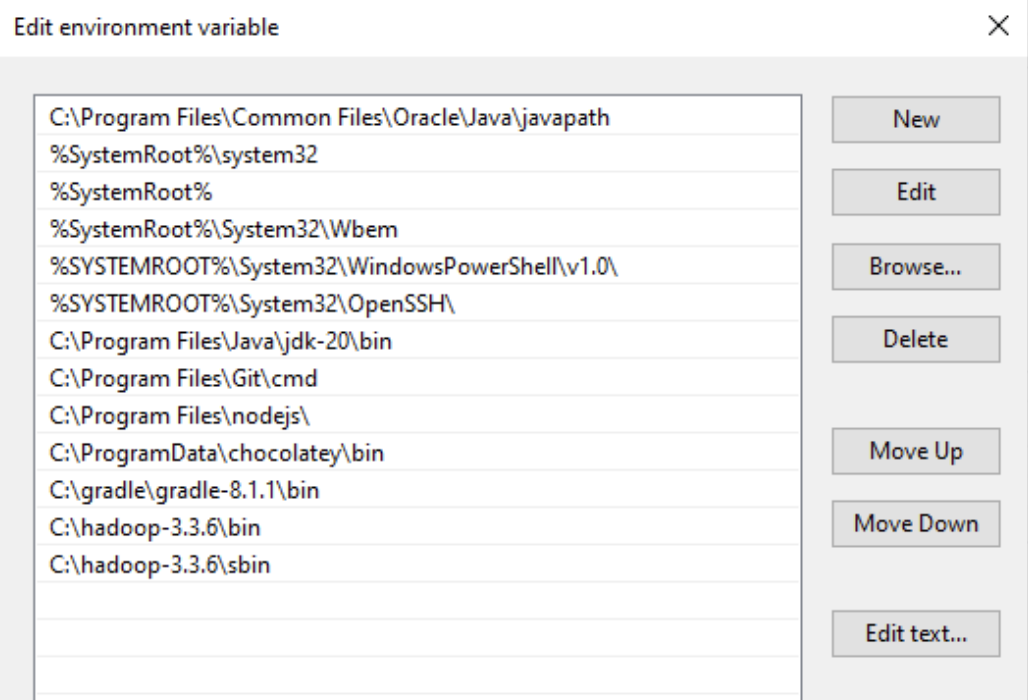
**Apache Hadoop in your local machine.**

A distributed database file system distributes and stores data across numerous nodes or servers, combining elements from distributed databases and file systems to provide scalable and dependable storage. Its architecture consists of nodes with storage and processing capabilities, central metadata servers that maintain file system structure, and data storage dispersed among nodes in blocks for redundancy and load balancing. Data replication provides durability and fault tolerance, while access control and security measures prevent illegal access. Load balancing prevents server overload, while data recovery techniques allow for restoration in the event of failure. Monitoring and management tools contribute to system health and performance.

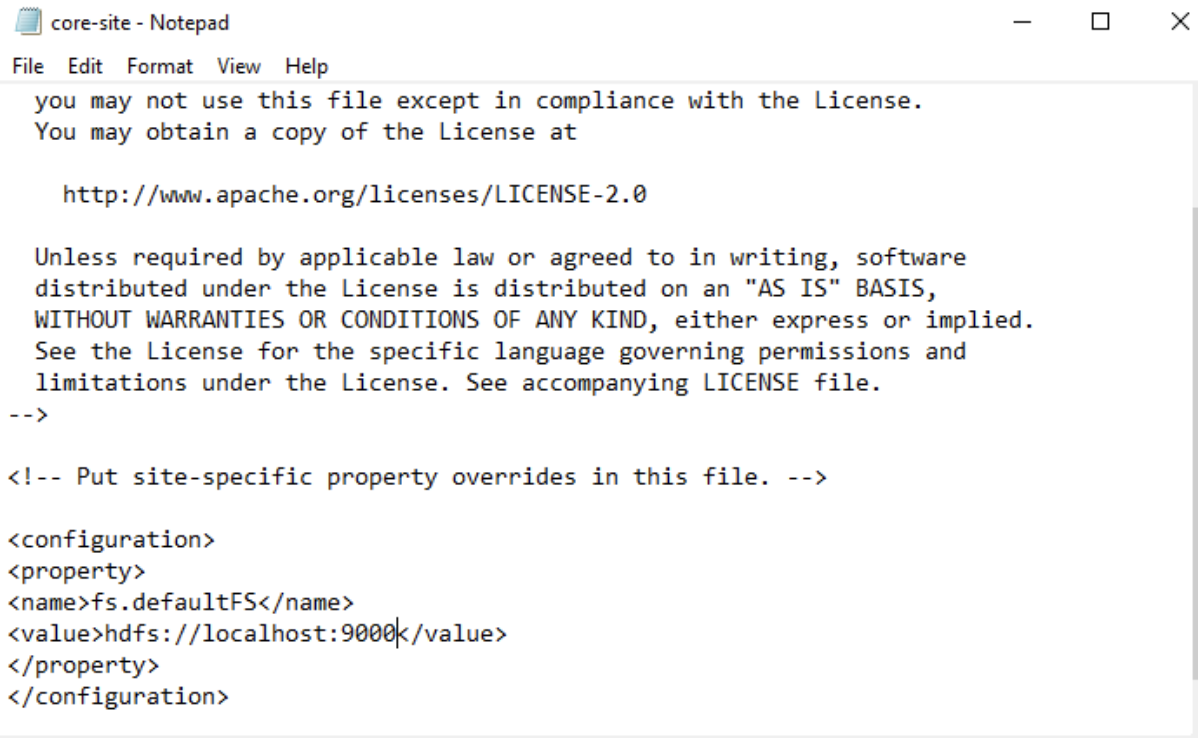
**Apache Hadoop Setup:**

1. First configure the Hadoop environment file by specifying the Java path. Then, add the Hadoop path to the environment variable.

****

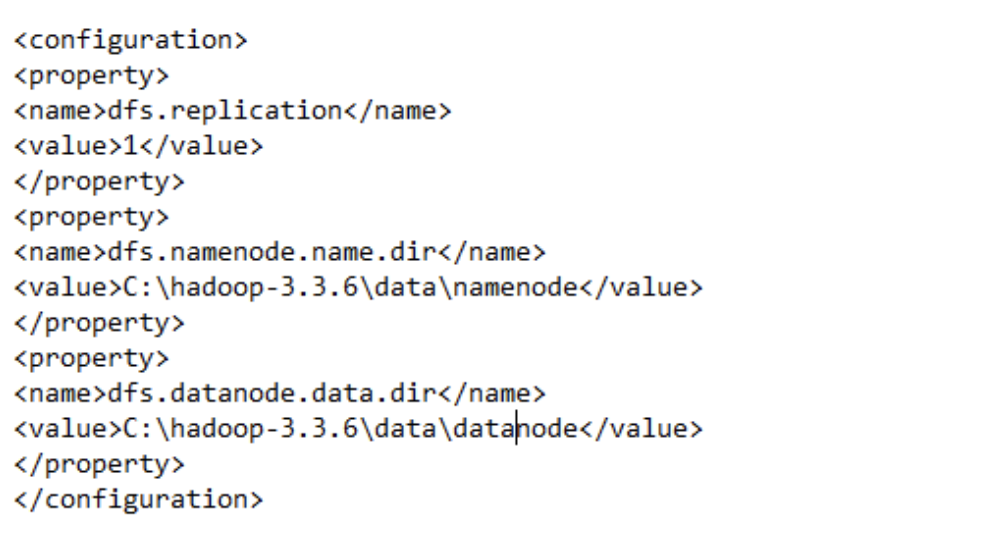
****

2. Configure core-site document

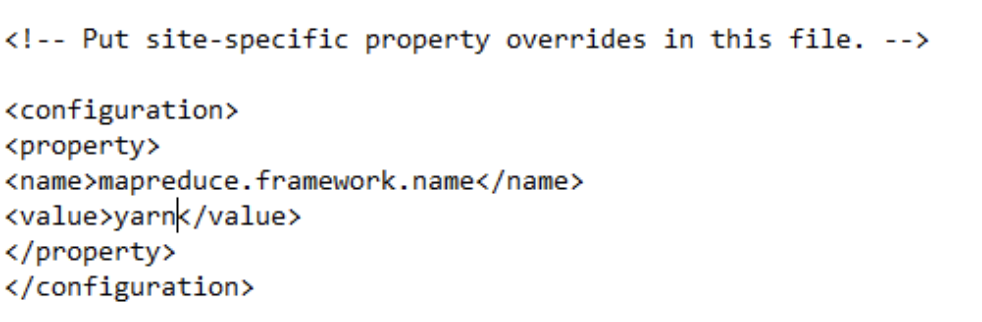
****

3. Now configure httpfs-site file or hdfs-site, either one will work.

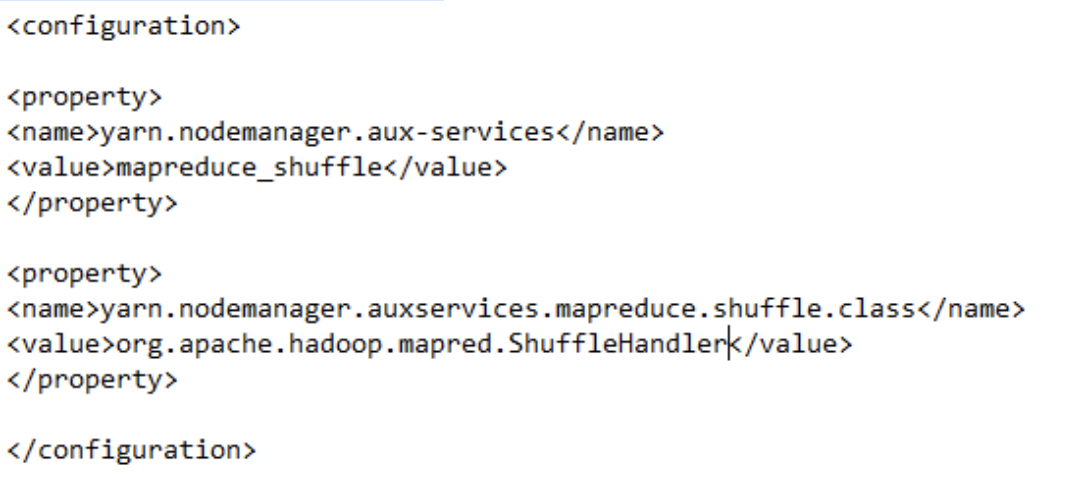
* 1. Create a folder named "data" and within it, create two subfolders: "namenode" and "datanode".
  2. Add the folder location to the configurable file httpfs-site or hdfs-site.



4. Now configure the mapred-site file



5. We need to configure yarn-site file



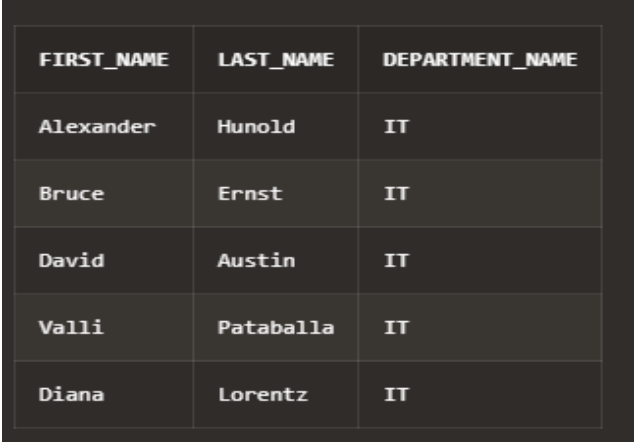
1. Format the hdfs namenode.
2. Start namenode and datanode by using command start-dfs.cmd
3. Start yarn by using command start-yarn.cmd
4. **Distributed Horizontal Fragmentation.**

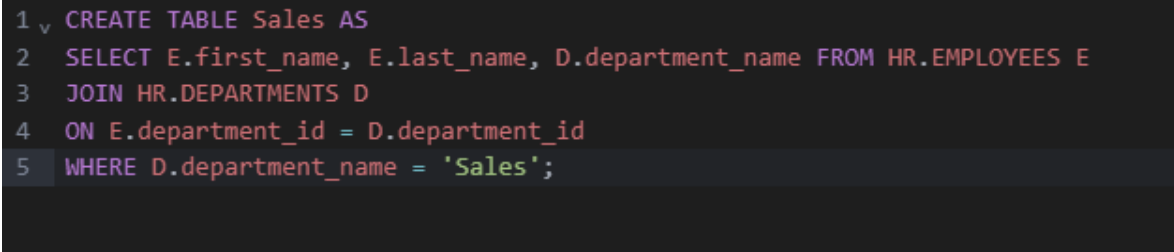
- Primary Horizontal Fragmentation

Divide Employees table with IT & Sales department in different fragments



**Output:**

****

****

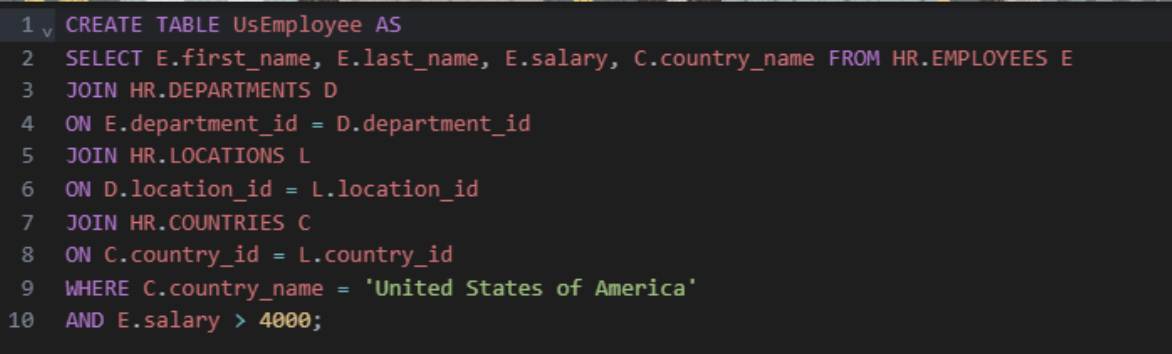
**Output:**

****

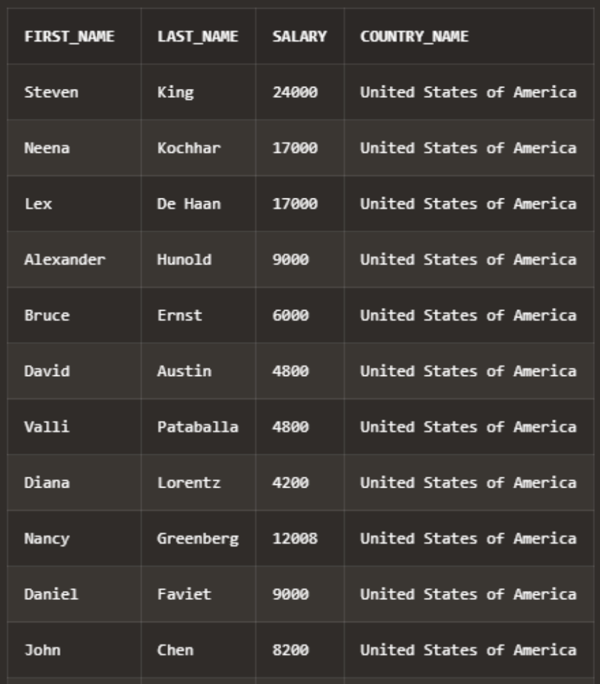
- Derived Horizontal Fragmentation

Create a different fragments for all employees who works on department located at

country 'United States of America' and have salary>4000



Output:

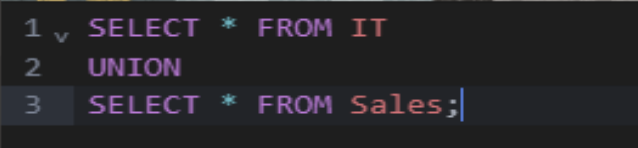


- Reconstruction from fragmented DB

- Fetch all employees who work on IT department and Sales department from

fragments created on section 1.

- Select all employees who are in IT and Sales department



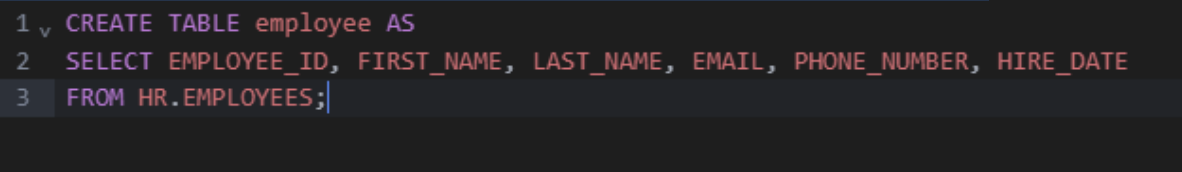
Output:



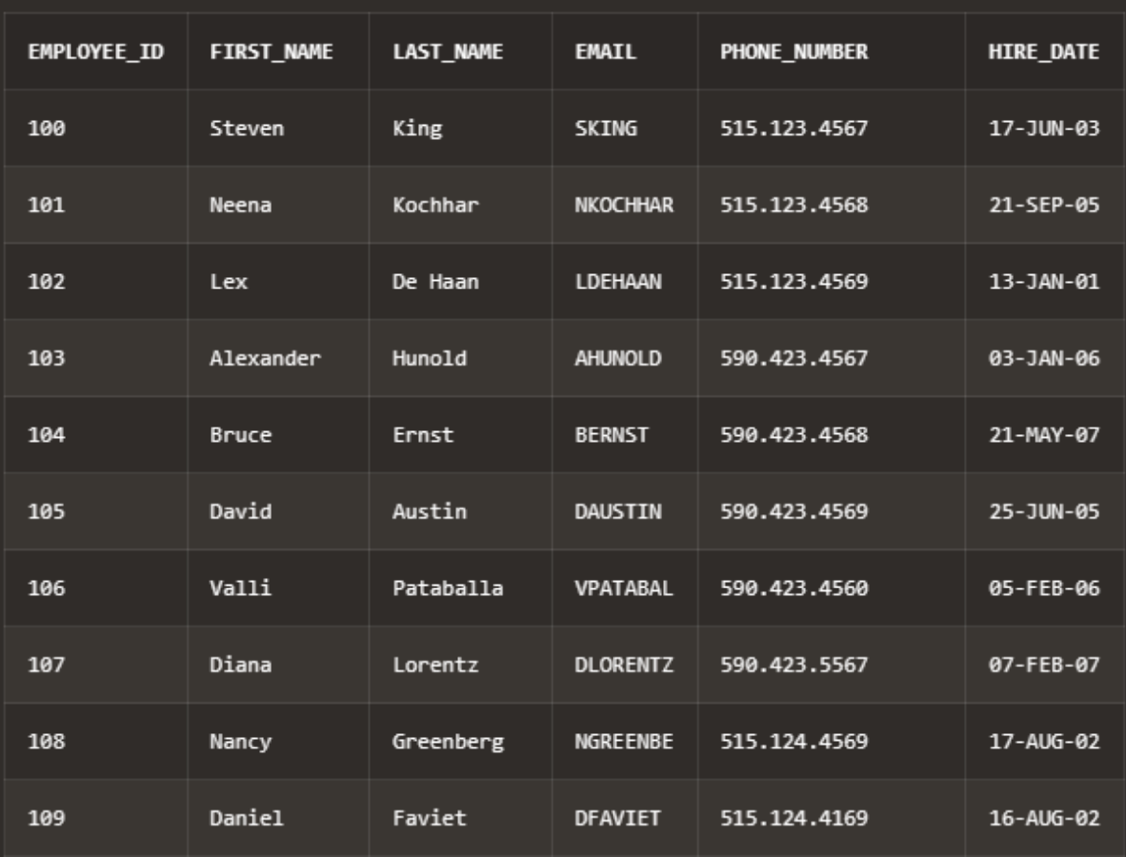
1. **Distributed Vertical Fragmentation.**

- Divide HR.EMPLOYEES table to hold employees related information in different

fragment and JOB description (all related columns) in different fragment.



Output:

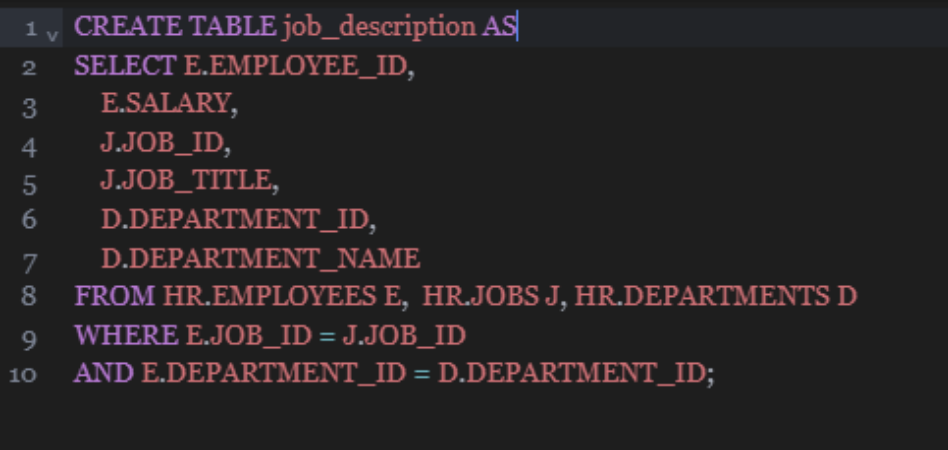


employee information must hold EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME,

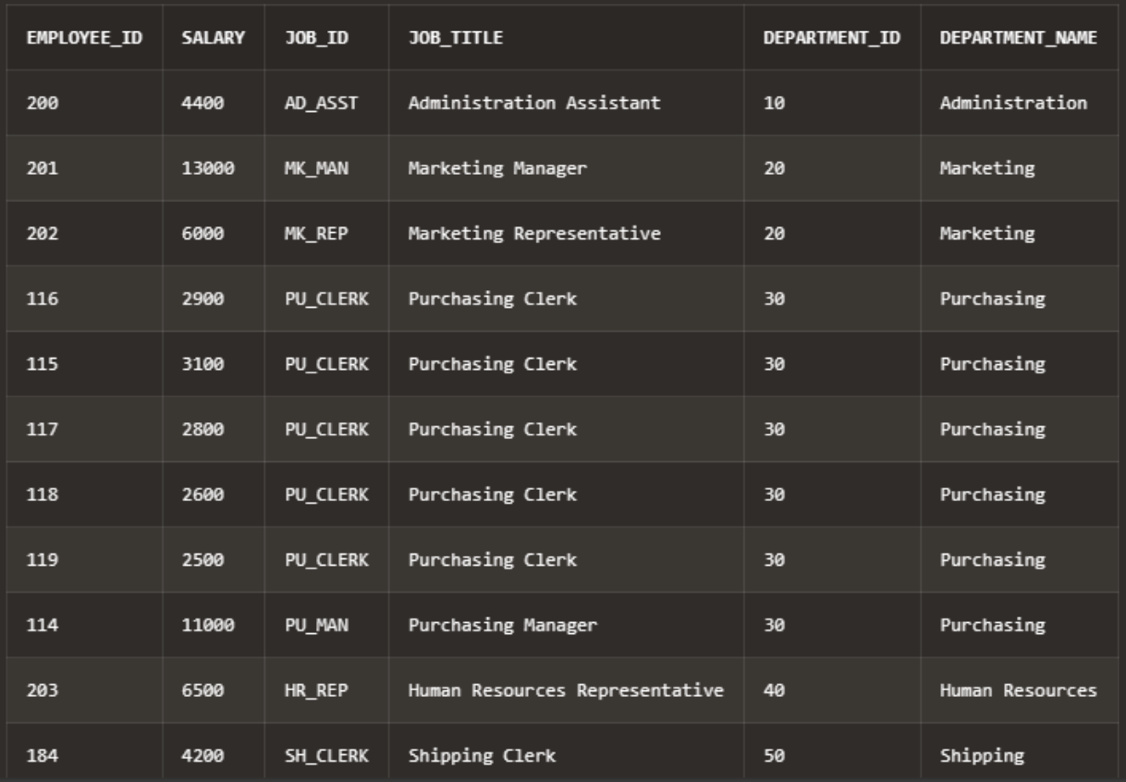
EMAIL, PHONE\_NUMBER,HIRE\_DATE

job description fragment must hold EMPLOYEE\_ID, JOB, SALARY,

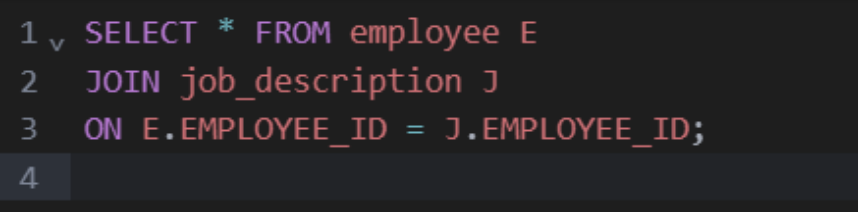
DEPARTMENT\_ID, DEPARTMENT\_NAME, JOB\_TITLE



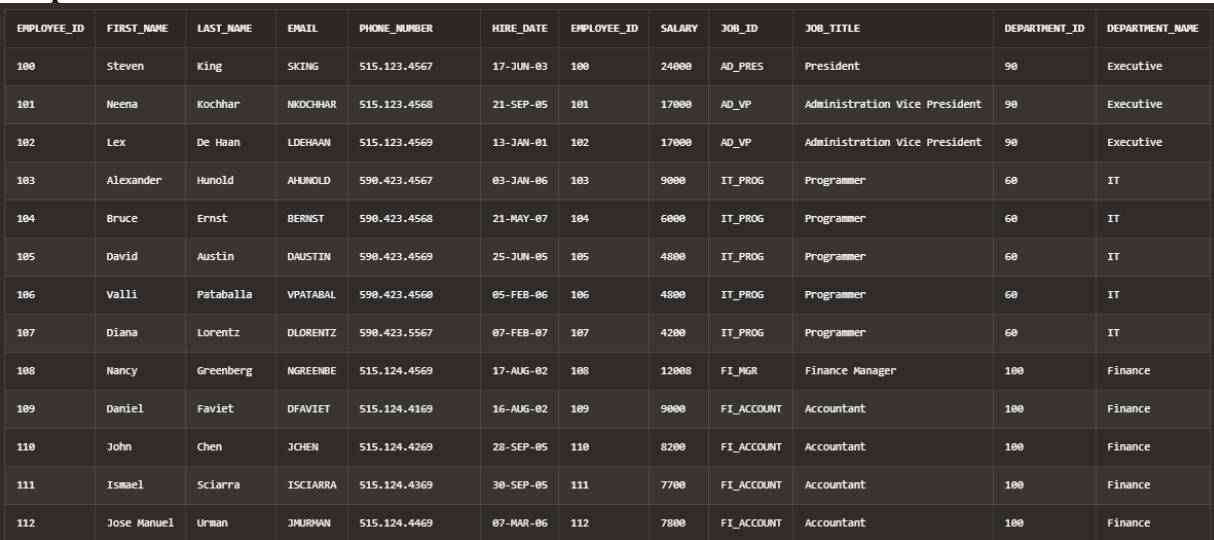
Output:



- Reconstruct all employees and job information from fragment tables.



Output:



1. **Using prolog, perform the following tasks for the Family Relations:**
2. Define facts to represent parent-child relationships in the family tree.
3. Define rules to represent the ancestor and descendant relationships.
4. Write queries to find all ancestors and descendants of a given person.
5. Define rules to represent sibling relationships.
6. Write queries to find all siblings.
7. Define rules to represent the aunt and uncle relationships.
8. Write queries to find all aunts and uncles of a given person.
9. Define a rule to represent the cousin relationship.
10. Write a query to find all cousins of a given person.