1. **What are relational databases? Describe, with the help of an example, how an RDBMS enables to relate data in one table to the data in another table.**

A relational database is a set of formally described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.

Elements of the relational database management system that overarch the basic relational database are so intrinsic to operations that it is hard to dissociate the two in practice.

The most basic RDBMS functions are related to create, read, update and delete operations, collectively known as [CRUD](https://searchdatamanagement.techtarget.com/definition/CRUD-cycle). They form the foundation of a well-organized system that promotes consistent treatment of data.

The most common means of data access for the RDBMS is via SQL. Its main language components comprise data manipulation language (DML) and data definition language (DDL) statements. Extensions are available for development efforts that pair SQL use with common programming languages, such as [COBOL](https://searchitoperations.techtarget.com/definition/COBOL-Common-Business-Oriented-Language) (Common Business-Oriented Language), [Java](https://www.theserverside.com/definition/Java) and [.NET](https://searchwindevelopment.techtarget.com/definition/NET).

1. **Why SQL is called a nonprocedural language?**

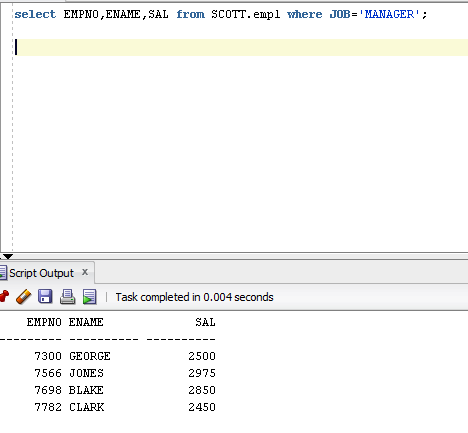
Non-Procedural means that SQL lacks the traditional control constructs such as IF-THEN-ELSE, WHILE, FOR, and GO TO statements found in procedural languages like C, Perl, Python, and other 3GL's.  
  
SQL is what is called a declarative language, meaning that you only need to specify what needs to be accomplished (such as a query or insert) and not how to do it; the DBMS determines and performs internally the step by step operation needed to obtain the result.

1. **Write down a brief history of SQL and explain why its standardization was necessary**

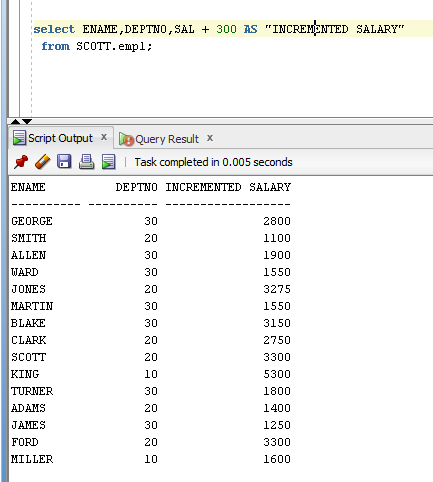
As we already know, prerelational databases did not have a set of commands to work with data. Every database either had its own proprietary language or used programs written in COBOL, C, and so on to manipulate records. Also, the databases were virtually inflexible and did not allow any internal structure changes without bringing them offline and rewriting tons of code. That worked more or less effectively until the end of the 1960s, when most computer applications were based strictly on batch processing (running from beginning to end without user interaction). Some benefits of standardization are as follows:

* Improves clarity — because a standard process will eliminate the need for guesswork or extra searching
* Guarantees quality — because work is done in a pre-defined, optimized way
* Promotes productivity — because your employees won’t need to ask around or comb documentation to get answers
* Boosts employee morale — because employees can take pride in having mastered the process and refined their skills
* Perfects customer service — because every ticket is handled in the best possible way

**Task 4:**



**Task 5:**



**TASK 6:**

