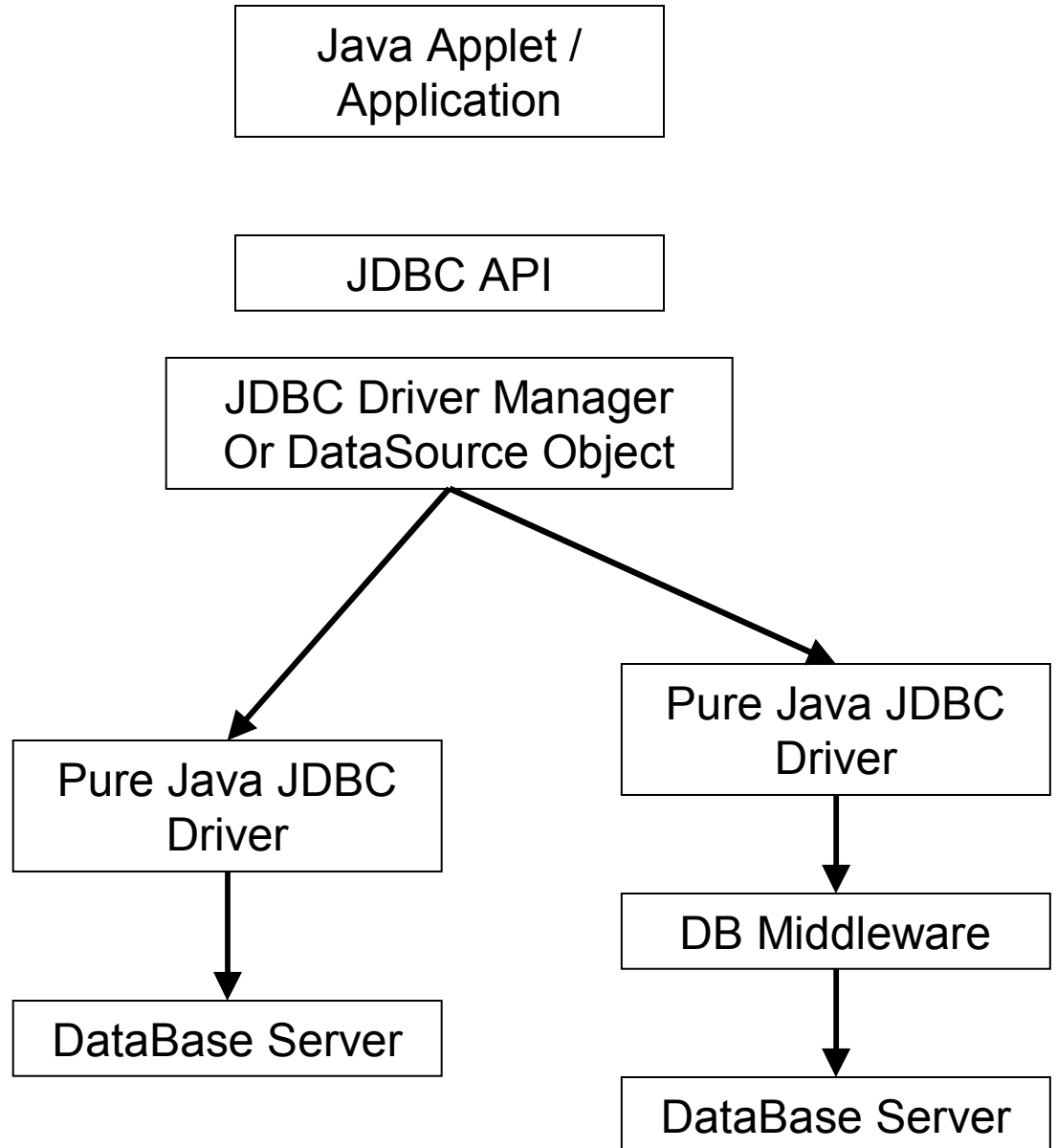
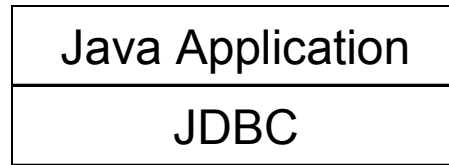


Java J2SE and J2EE  
SDKs provide the JDBC  
Java DataBase  
Connection API  
in the java.sql package.



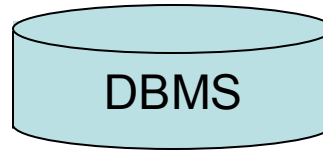
2 tier architecture



Client machine

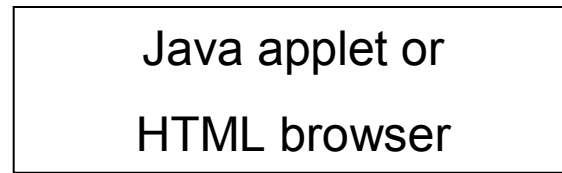


DBMS proprietary protocol



Database server

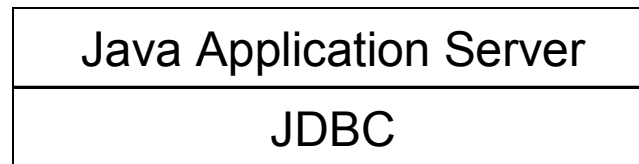
3 tier architecture



Client machine (GUI)



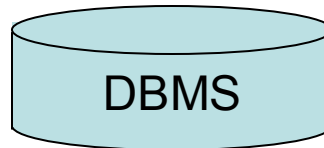
HTTP, RMI, CORBA, ... calls



Server machine  
(business logic)



DBMS proprietary protocol

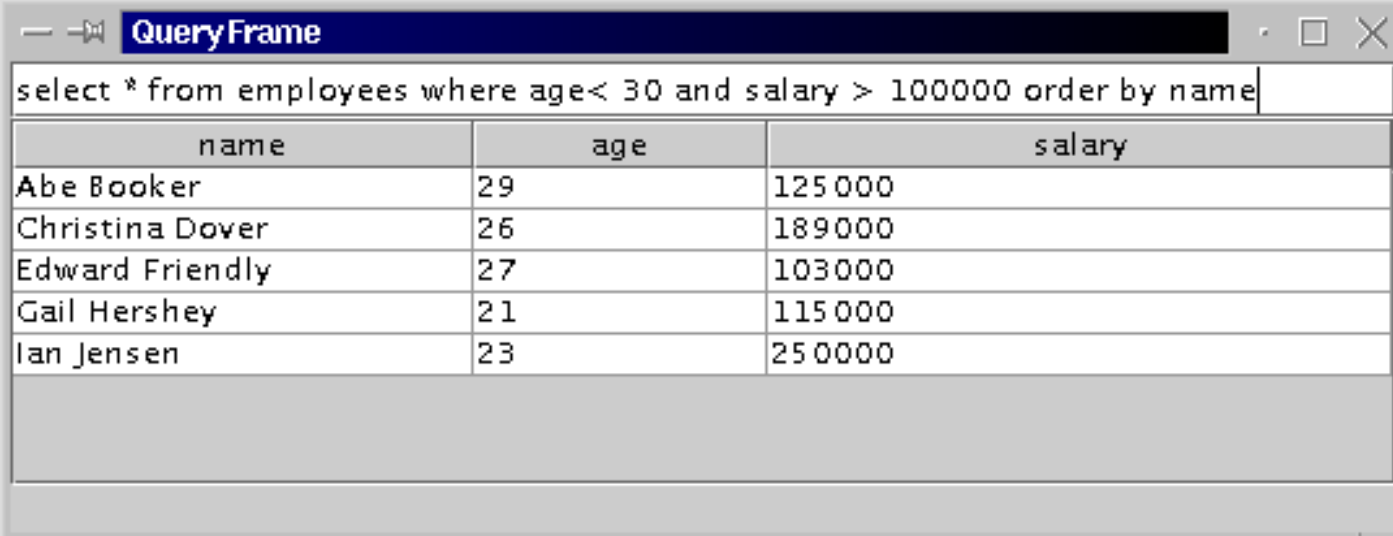


Database server

The `java.sql` package defines the JDBC API for sending SQL queries and update statements to a database, and for obtaining query results. This example demonstrates how to use JDBC to connect to a database and obtain a `java.sql.Connection` object, how to execute a SQL query using a `java.sql.Statement` object, and how to work with the query results returned in a `java.sql.ResultSet` object.

`JTable` can display any type of tabular data as long as it implements the `javax.swing.table.TableModel` interface. The examples in this article include a `TableModel` implementation that interprets a JDBC `ResultSet` object, so that the queried data can be displayed by a `Jtable`.

[http://www.oreilynet.com/pub/a/oreilly/java/news/javaex\\_1000.html](http://www.oreilynet.com/pub/a/oreilly/java/news/javaex_1000.html)



The screenshot shows a Java Swing window titled "QueryFrame". The window contains a text area with the SQL query: `select * from employees where age < 30 and salary > 100000 order by name`. Below the text area is a table with three columns: "name", "age", and "salary". The table contains five rows of data:

name	age	salary
Abe Booker	29	125 000
Christina Dover	26	189 000
Edward Friendly	27	103 000
Gail Hershey	21	115 000
Ian Jensen	23	25 000