QUESTIONNAIRE ON THE OBJECT-RELATIONAL IMPEDANCE MISMATCH

Name: Marc Duran López

Name: Ferran González García

Question 1. Consider the concepts of Complex Concept (and all their related concepts, such as Type Classes, Extensibility, etc.) and Rich Type System. Are they equivalent, compatible or do they represent a conflict between both approaches? Justify your answer

Pensamos que ambos sistemas son **compatibles**. Esto es debido a que ambos buscan una solución a través de una óptica similar que són los constructores. El Rich Type System permite que haya nuevos constructores con el abstract data type y que se puede llegar a generalizar. Es por eso, y porque en el manifesto RDB, se explica que se podría llegar a hacer con RDS cualquier tipo de constructor propuesto en OODB.

Question 2. Consider the concepts of Object Identity and PK (include in the discussion the concepts of reference and FK). Are they equivalent, compatible or they represent a conflict between both approaches? Justify your answer

Consideramos que los sistemas son **conflictivos** en este aspecto. La PK tiene un valor semántico con significado mientras que el OID es un valor único por definición y sin significado. Pesé a que el modelo que presenta RDB puede llegar a tener una aplicación con OID, OODB plantea que esto llevaría más problemas con la verificación de la integridad y otros procesos como el *garbage collector*.

Question 3. Consider how each paradigm access data. That is, high-level programming languages (typically computationally complete) Vs. SQL and the proposed extensions (i.e., constraints enforcement, updatable views, interfaces, etc.). Are they equivalent, compatible or they represent a conflict between both approaches? Justify your answer

Después de leer ambos manifiestos, pensamos que los dos sistemas son **equivalentes**. Ambos buscan tener lenguajes declarativos para intentar hacer la búsqueda en base de datos lo más sencilla posible.

Question 4. Consider the position of both paradigms with regard to (code) encapsulation. Are they equivalent, compatible or they represent a conflict between both approaches? Justify your answer

Pensamos que ambos sistemas son **compatibles**. Esto es debido a que RDB propone el sistema de funciones como un punto de entrada para los usuarios, en vez de crear ellos mismos las propias sentencias SQL, mientras que el sistema de OODB propone no solo guardar el objeto en la base de datos, sino también las funciones/operaciones adecuadas. Pasar de un sistema a otro se podría llevar a cabo y consideramos que es relativamente fácil.

Question 5. Consider the concepts persistency, concurrency and recovery. Are they equivalent, compatible or they represent a conflict between both approaches? Justify your answer

Por lo que hemos consultado en los textos, ambos sistemas son **equivalentes**. Esto se debe principalmente a que en el manifiesto de OOBD se especifica en cada una de las categorías mencionadas que se intenta replicar el mismo sistema ya implementado, es decir, el de RDB.

Question 6. Consider the concepts of Inheritance in both paradigms. Are they equivalent, compatible or they represent a conflict between both approaches? Justify your answer

Consideramos que ambos sistemas son **compatibles**. Esto es porque ambos proponen la existencia de superclases como “Persona” que agrupa tanto a “Estudiante” como “Empleado”, pese a las posibles restricciones de edad que puedan presentar. Ambos proponen la misma solución. La principal diferencia la encontramos sobre la multiherencia, en la cual en el manifiesto de OOBD parece que no se llegó a consenso dentro de la comunidad, mientras que RDB defiende la necesidad de esta característica.

Question 7. Consider the concept of open system and the position of each paradigm. Are they equivalent, compatible or do they represent a conflict between both approaches? Justify your answer

Los dos sistemas comparten algunos principios comunes relacionados con la apertura y la independencia del lenguaje, pero no son equivalentes en su totalidad. El Manifiesto de los Sistemas de OOBD se enfoca de manera más estrecha en la tecnología de bases de datos orientadas a objetos, mientras que el Manifiesto de los Sistemas de RDB tiene un alcance más amplio, que abarca varios aspectos del diseño de sistemas de bases de datos y la apertura.

Por tanto, en este aspecto, los dos sistemas pueden considerarse **compatibles** siempre que se implementen de manera que cumplan con los principios de sistemas abiertos y la independencia del lenguaje. Aunque provienen de enfoques y contextos diferentes, no

necesariamente entran en conflicto si se diseñan y desarrollan con la apertura y la interoperabilidad en mente.

La compatibilidad dependerá de cómo se apliquen estos principios en la práctica dentro de cada paradigma y de su capacidad para funcionar en entornos heterogéneos e intercambiar datos con otros sistemas sin restricciones indebidas.

Question 8. Why do you think object-oriented databases did not succeed? In other words, what was the main drawback of object-oriented databases?

El principal inconveniente de las bases de datos orientadas a objetos fue la falta de estandarización, lo que llevó a problemas relacionados con la complejidad, el rendimiento, la integración y el apoyo de la industria. Estos desafíos, combinados con la dominancia de las bases de datos relacionales y la aparición de soluciones objeto-relacionales, contribuyeron al éxito limitado de las OODBs.

Question 9. What are the standard object-oriented features introduced by object-relational databases to smooth the impedance mismatch with object-oriented programming languages?

Algunas de las características añadidas en los Sistemas de Bases de Datos Relacionales para evitar un sobrecoste de traducción podría ser la posibilidad que se ofrece para que el usuario pueda definir sus propios tipos de datos, como se ve en el apartado del *Rich Type System*. Otra de las características a tener en cuenta es la herencia, una funcionalidad completamente implementada con la visión de los lenguajes de la época que iniciaban en la orientación a objetos. Finalmente, encontramos la encapsulación, como se menciona en la respuesta a la pregunta 4.

Question 10. Consider a Java application that defines tones of classes and stores its objects in a PostgreSQL by means of JDBC. Consider the same application but now running DB4o (or any other object-oriented database supporting Java) in the backend. When executing the mappings to store the Java objects into the database, could you name two OO features whose mapping to PostgreSQL (i.e., the JDBC code lines executed) will clearly perform worse than its counterpart mapping to DB4o? Justify your answer

Por la propia naturaleza de ambos sistemas, encontramos que DB4o tendrá un mejor rendimiento en cuanto a la manera de guardar los datos que tengan múltiples relaciones entre las clases. Esto se debe a que el sistema no relacional, gracias a su sistema de Identificación de Objetos, es capaz de almacenar más fácilmente las

referencias a otros objetos ocupando menos espacio, favoreciendo también las operaciones de asignación y copias.

Por otra parte, la herencia podría llegar a suponer un problema. Mientras que las bases de datos no relacionales proporcionan un sistema nativo para las clases con una estructura muy moldeable a las necesidades del usuario, el *mapping* que requiere la traducción a la RDBMs puede suponer un sobrecoste importante.