Proxecto

1.- Obxectivo

O obxectivo académico de este traballo é a demostración de competencia básica no uso dos distintos tipos de solucións tecnolóxicas que se traballaron nas prácticas da materia. Para acadar este obxectivo, o alumnado ten que atopar un conxunto de datos dentro dunha temática asignada, transformar os datos para poder ser importados nas distintas tecnoloxías de xestión de datos e resolver un pequeno conxunto de consultas de dificultade variada. As tecnoloxías consideradas son as seguintes:

1.- Base de datos PostgreSQL con un modelo en primeira forma normal.

2.- Base de datos PostgreSQL con un modelo complexo, que soporte tipos de datos compostos e arrays.

3.- Base de datos PostgreSQL con tipo de datos JSON para soportar estruturas complexas.

4.- Base de datos PostgreSQL con arquitectura distribuída (CiTUS Data)

5.- Base de datos NoSQL con modelo documental MongoDB

6.- Base de datos NoSQL con modelo de grafos Neo4J

7.- Base de datos NoSQL con modelo column-Family (HBase) [Opcional]

2.- Grupo de traballo e selección da temática

O alumnado traballará en **grupos de 3** (excepcionalmente poderán admitirse algúns grupos de 2 si existe unha xustificación clara). Cada grupo de traballo fará unha descrición curta dun ámbito de aplicación no que é preciso almacenar e consultar datos. Esta descrición debe incluír referencias a conxuntos de datos relacionados con esa temática. O profesorado repartirá estas descricións entre os grupos de traballo, de maneira que ningún grupo realice o traballo sobre o ámbito que propuxo.

3.- Modelo relacional

Utilizando como base a descrición da temática proporcionada, o grupo de traballo localizará os conxuntos de datos necesarios para preparar o contido dunha base de datos que debe de ter cando menos os seguintes elementos:

- Unha táboa que represente unha entidade con un atributo de texto dentro do que sexa necesario facer procuras por palabra clave.
- Un mínimo de tres táboas en total, relacionadas entre si, con cando menos dúas relaciones de tipo un a varios ou varios a varios.
- Unha relación reflexiva, e dicir, unha relación entre unha entidade e si mesma.
- Opcionalmente, poden incluírse atributos que almacenen representacións xeométricas das entidades.

Para conseguir o contido será posible combinar datos obtidos de un ou varios conxuntos de datos con datos xerados de forma manual ou automática.

Deben propoñerse entre 5 e 8 necesidades de información que requiran consultas SQL que involucren distintas partes da sintaxe da linguaxe.

1.- Cando menos unha consulta debe precisar do JOIN entre como mínimo dúas táboas.

2.- Cando menos unha consulta debe precisar do uso de funcións de agregado.

3.- Cando menos unha consulta debe precisar do uso da cláusula GROUP BY

4.- Cando menos unha consulta debe de precisar do uso da cláusula HAVING

5.- Cando menos unha consulta debe de precisar do uso da cláusula UNION

7.- Cando menos unha consulta debe de requirir a procura baseada en palabras clave sobre campos de texto.

8.- Cando menos unha consulta debe de precisar a navegación varias veces (mais de dúas) a través da relación reflexiva.

9.- Opcionalmente, pode incluírse algunha consulta espacial sobre as representacións xeométricas.

Cada unha das necesidades de información deben de ser resoltas con código SQL que debe de ser explicado.

4.- Modelo relacional estendido

Crearase un novo esquema dentro da base de datos para gardar os mesmos datos usando alternativas de modelado relacional que incorporen estruturas de agregación. En concreto, realizaranse as dúas tarefas seguintes.

Creación de tipos de datos compostos necesarios para almacenar os datos da base de datos no número mínimo de táboas, usando estruturas de agregación nativas de PostgreSQL. Debe decidirse en que dirección se agregan os datos para conseguir isto. Despois de realizar as consultas da sección anterior sobre esta nova base de datos, ademais de explicar o código, debe de explicarse cales das consultas se ven beneficiadas polo orde de agregación e cales se verían prexudicadas.

Utilización do tipo de datos JSON para crear unha nova versión da base de datos con agregados, que de novo utilice o mínimo número de táboas. Utiliza unha dirección de agregación distinta a utilizada antes e explica cales serían agora as consultas beneficiadas e cales serían as prexudicadas.

Deben compararse os tempos de execución das consultas con: i) o modelo relacional, ii) o modelo con agreados nativos de PostgreSQL e iii) o modelo con agregados implementados en JSON.

5.- Base de datos relacional distribuída

Nesta parte do proxecto executaranse as consultas do modelo relacional sobre un cluster. Crearase un cluster de 3 máquinas, a máquina GreiBD utilizarase como coordinador de CITUS DATA. Dous clones da máquina GreiBDServerBase servirán como workers. Unha vez creado o cluster, importaremos os datos en formato relacional. Distribuiranse os datos entre os workers de forma axeitada de maneira que poidan resolverse as consultas propostas para a sección 3. Debe decidirse cales táboas deben de ser distribuídas e por que campo, e que táboas deben de

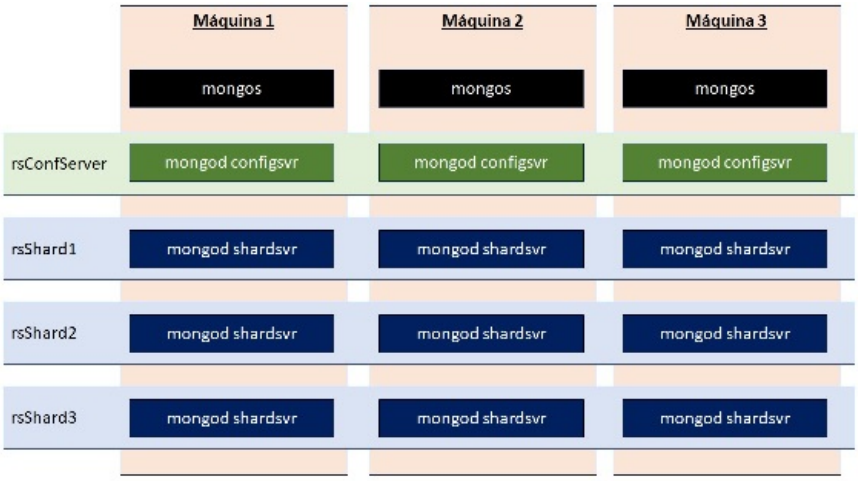
ser replicadas en todos os nodos. Analizarase o impacto que ten sobre as consultas a parada de un dos dous nodos worker.

Comprobaráse agora o impacto que ten a parada de un nodo worker si utilizamos un factor de replicación 2. Para isto, buscaremos na web de CITUS información sobre como incluír replicación durante o particionamento. Utilizaremos a variable citus.shard_replication_factor para definir un factor de replicación 2. Para comprobar o impacto de este factor de replicación deberáse desfacer a distribución das táboas (usando undistribute_table()) e volver a distribuílas, ou alternativamente borrar as táboas, volver a importalas e despois distribuílas.

Por último, deberase buscar información sobre o uso do formato columnar para as táboas de CITUS. Despois de borrar de novo as táboas, debemos importar de novo as mesmas, pero agora usando este formato columnar na táboa ou táboas que consideres oportuno. Comproba o impacto no tempo de resposta das consultas o uso de este formato columnar.

6.- Base de datos NoSQL documental: MongoDB

Debe crearse un cluster de MongoDB seguindo a arquitectura que podes ver na imaxe de abaixo.



Pódese usar GreiBD e dous clones de GreiBDServerBase. Importa os datos da túa base de datos usando agregación e o número mínimo de coleccións JSON. Elixe a dirección de agregación e explica cales son as consultas beneficiadas e cales as prexudicadas. Indexa e particiona a colección (ou coleccións) da maneira que creas mais axeitada e despois resolve as consultas propostas na sección 3. Comproba o que ocorre coa execución das consultas se paras unha das máquinas. Fai a comprobación de novo parando dúas das máquinas. Fai probas con distintas opcións de preferencias e compromisos de lectura para ver o impacto que ten no comportamento das consultas cando tes paradas dúas das tres máquinas.

7.- Base de datos NoSQL de grafos: Neo4J

Deberá deseñarse un modelo de grafos para representar a información da base de datos utilizada nas seccións anteriores. Importaremos os datos a este modelo de datos en Neo4J directamente dende PostgreSQL usando unha conexión JDBC. Resolveremos as consultas propostas na sección 3. Explica as vantaxes que ten esta solución para a implementación da consulta que precisa navegar a través da relación reflexiva, respecto ao uso de tecnoloxías relacional.

8.- Base de datos NoSQL de tipo Column Family: HBase [OPCIONAL]

Debe deseñarse un modelo de tipo Column Family para representar a información da base de datos utilizada nas sección anteriores. Implementaráse unha aplicación JAVA que permita importar os datos e executar as consultas propostas na sección 3. Debes de delegar o máximo de funcionalidade posible en HBase cando implementes as consultas.

6.- Documentación a entregar e prazo de entrega

Para a entrega do proxecto será necesario xerar a seguinte documentación.

- Un **arquivo PDF** con explicacións sobre a realización de cada unha das tarefas. Débese explicar co suficiente nivel de detalle todo o código utilizado para resolver os problemas (consultas, comandos, algoritmos, etc.). Deben incluírse os resultados obtidos en cada paso. Si algún resultado é moi grande, pode incluírse unha parte pequena a modo de exemplo, indicando o seu tamaño. Ademais débense incluír todas as demais explicacións requiridas en cada sección. Debe realizarse unha maquetación apropiada do documento para facilitar a súa lectura.
- Unha copia da **base de datos** relacional producida na sección 1. Debe incluírse un script para a creación das táboas e un arquivo csv co contido de cada táboa.
- Todos os **arquivos de texto** que se consideren necesarios que conteñan todo o código utilizado para resolver os problemas, incluíndo consultas nas distintas linguaxes, comandos executados para iniciar procesos e código de algoritmos. O obxectivo de estes arquivos é facilitar a proba das solucións por parte do profesorado, usando os datos dos arquivos do punto anterior.

Todos os arquivos deben de xuntarse nunha carpeta que debe de ser comprimida usando zip e subida ao elemento correspondente do campus virtual antes do día **22 de decembro de 2024 as 23:59**.

7.- Criterios de avaliación

A execución correcta e completa da parte obrigatoria do proxecto permitirá alcanzar a puntuación máxima asignada no programa da materia. A parte opcional permitirá, en caso de ser completa e correcta, alcanzar como máximo un 10% adicional de puntuación, que poderá ser utilizada para compensar problemas de completitude ou corrección na parte obrigatoria.

Para analizar a completitude analizaranse os seguintes aspectos:

- Tamaño e complexidade do conxunto de datos utilizado
- Cantidade e complexidade das necesidades de información propostas
- Cantidade de tarefas resoltas.

Para analizar a corrección das solucións teranse en conta os seguintes aspectos:

- Eficacia do código xerado (responde o código as necesidades previstas?)
- Eficiencia do código xerado (funciona o código cun tempo de execución e cun consumo de memoria razoable?)
- Coherencia dos resultados obtidos.
- Calidade das explicacións incluídas no PDF.
- Calidade da redacción e da presentación do documento.