Sid

1

* 1. Esquema relacional (origem)

Todas as tabelas mais uma por cada, neste caso 16 tabelas, em que 8 são tabelas logs com campos adicionais (utilizador, data, operação, id)

Exemplo:

Tabela variáveis

|  |
| --- |
| Variável |
| IdVariavel |
| NomeVariavel |

Tabela logo variáveis

|  |
| --- |
| VariávelLog |
| IdVariavel |
| NomeVariavel |
| Utilizador |
| Data e Hora |
| Operação |
| Id \* |

\* É realmente necessário ter este ID?? Já temos o IDVariável que pode ser usado como chave! Não percebo o porque deste ID?! Para a mesma variável, ou seja, o mesmo id podes ter várias operações um insert e um select pelo que precisas de outro id para diferenciar.

Tabelas logs vão servir para passar informação para o auditor ver onde tem a informação da operação feita (select, insert, update, delete) qual o utilizador que fez essa operação e quando a efectuou.

Tem que ser discutido quando existe restrict, cascate ou set null na operação delete. No update custuma ser sempre cascade. ( se optarmos por escolher cascade, temos se saber argumentar bem o porquê desta decisão e não pode ser só porque “costuma ser“). No ano passado o grupo com quem defende mos o projecto não pôs cascade no update e o prof disse que não fazia sentido que mesmo ele nunca teve um caso em que não fosse cascade no update que não havia muito a dizer sobre isso é o mais óbvio fazes update e altera tudo porque até nem vai influenciar em nada ser tudo alterado.

1.2 - Utilizadores Base de Dados de Origem

Admin (DBA), admin(aplicação), investigador

Não concordo como facto de haver dois admin. Esta opção apenas trará confusão e complexidade ao sistema. Se virmos o Use Case apenas refere AdministradorAplicação, que faz manutenção de dados. O admin aplicação é realmente o que o prof pede mas qualquer projecto tem um admin DBA que tem permissão para tudo. Aliás no use case o admin aplicação não tem permissão para tudo o que não faz sentido haver admin que não possa fazer tudo daí haver o de aplicação e o dba que “manda” o admin DBA inclusive vai ter permissão para criar admins aplicação (pode haver mais que um).

Investigador pode aceder as tabelas cultura, variaveisMedidas, medições, medicoesLuminosidade, medicoesTemperarura. Nestas tabelas existe tabelas em que pode fazer tudo outras que apenas pode fazer por exemplo select temos que definir isso. Nas tabelas cultura apenas podem fazer operações nas suas culturas. Na tabela medições ocorre o mesmo.

De acordo com o que interpreto do UML da BD e do Use Case, penso que o Investigador apenas pode aceder à tabela de Medições, e pode fazer insert, update e select nesta tabela. Se reparamos bem, a tabela da cultura é uma composição, ou seja, esta já tem as chaves de Variáveis e Cultura. Com esta arquitetura não existe a possibilidade de um investigador inserir ou alterar o nome de uma cultura, nem de alterar ou inserir as Variáveis. Torna o sistema mais rígido, mas ao mesmo tempo mais seguro e controlado. Mas na minha opinião, deve ser dada a possibilidade ao Investigador para realizar apenas select à tabela de Variáveis, Cultura e Variáveis Medidas, para que este tenha acesso aos Ids e Nomes, por exe:

O investigador apenas pode fazer select na cultura e variáveis. Podendo fazer todas as operações nas medições. Não tenho a certeza mas até no left Join acho que tens que ter permissão para select no exemplo abaixo tás a querer retornar o nome ou seja um campo que não existe na tabela medições. O ano passado o prof aceitou o investigador aceder a outras tabelas mas apenas com select. Mas isso é sempre possível discutir em grupo

Select Variáveis.NomeVariavel, Cultura.NomeCultura, Medições.ValorMedição

From Medições

Left join Variáveis on Variáveis.IDVariável = Medições.IDVariável

Left join Cultura on Cultura.IDCultura = Medições.IDCultura;

Este select permitiria ao investigador obter o nome da cultura, da variável e da medição sem ter de saber os Ids

Acho que temos também de criar Stored Procedures que permitam ao investigador apenas aceder ás suas culturas. Exato por isso é que ele acede também a tabela cultura mas só para a operação select.

Quanto às tabelas MediçoesLuminosidade e MediçõesTemperatura segundo o que percebi na aula e o que está no Use Case, estas tabelas são apenas para registo dos sensores. Podemos dar a possibilidade ao investigador de realizar select às mesmas, mas não deverá ter a possibilidade de alterar ou inserir, senão poderá alterar e comprometer os valores medidos pelos sensores. Sim concordo.

Assim, com esta arquitetura, apenas o administrador poderá inserir ou alterar novas culturas ou variáveis.

Admin(aplicação) serve para não sobrecarregar o admin(DBA), acede a tudo menos a tabela medições que pertence apenas ao investigador. Igual ao investigador a tabelas que apenas pode fazer por exemplo select temos também que definir isso.

Tal como disse anteriormente, não concordo nem percebo a vantagem. O que significa sobrecarregar neste caso?? O Use Case apenas refere um admin. Existe sempre um admin que manda em tudo que o prof não refere mesmo para ver se chegamos lá. E um admin que é o que ele refere.

Admin(DBA) pode fazer tudo.

1.3 – Gestão de Logs

1.3.1 – Triggers de suporte à criação de logs Base de Dados de Origem

3 Triggers por cada tabela para o insert, delete, update. A função do trigger é colocar informações nas tabelas logs, ou seja, quando o investigador inserir uma medição na tabela medições o trigger insert da tabela medições vai por essa informação na tabela medicoesLog pondo o campo utilizador com o nome do investigador a operação como insert e a data dessa operação.

Apenas penso que deveria haver Data e Hora no mesmo campo ou em campos separados. Mas uma tabela de LOG deve ter sempre horas. Se o mesmo utilizador consultar/alterar 2 ou mais vezes a mesma tabela no mesmo dia, apenas terás um registo se só registarmos a Data. Sim é possível considerar timestamp que tem tanto a data como a hora. Provavelmente é o que o prof pretende.

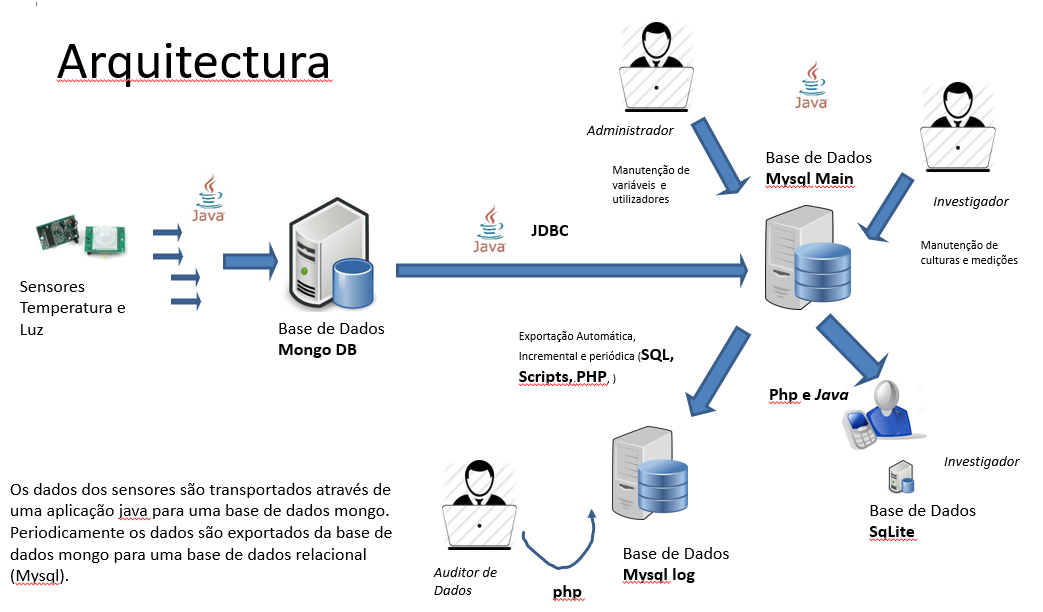
1.3.2 – Stored Procedures de suporte à criação de logs (se relevante)

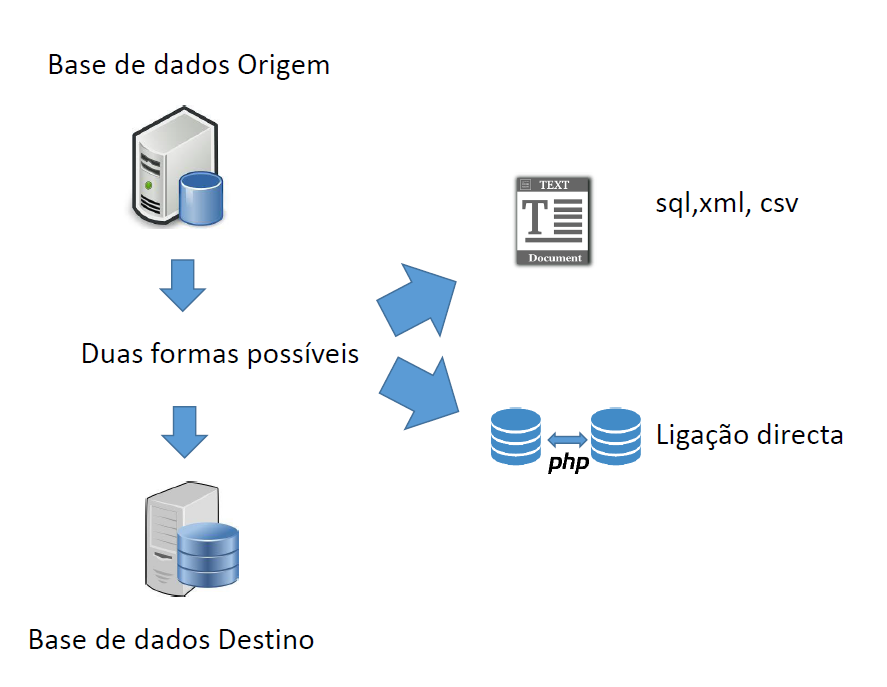
4 por cada tabela para o select, insert, delete, update. A função dos SP é para facilitar as operações.

Exemplo:

Em vez de fazer: insert into cultura values(..,..,..) basta fazer call insertCultura(…,…,…)

Neste ponto temos de ter atenção aqui é para colocar apenas SP na BD de origem e apenas para as tabelas de Logs. Não os sp funcionam para inserir nas tabelas reais os Triggers e que adicionam nas tabelas logs. Nenhum utilizador faz operações nas tabelas logs. Só o auditor e só pode fazer select e isso é na tabela destino que como referiste não entra aqui. Os sp deste ponto e para as tabelas reais.





1.4

1.4.1 – Esquema relacional (destino)

Por ficheiro ou PHP dependo do que nos vai calhar. Por ficheiro acho que temos que fazer um evento que vai exportar a informação dos logs para os ficheiros, neste caso um ficheiro para cada tabela log e mais outro evento para ler o ficheiro para a base dados do auditor. Por PHP não sei como funciona.

Aqui será apenas para apresentar o esquema relacional da BD Mysql Log do auditor

1.4.2 – Forma de migração

Programas usados e breve explicação de como a migração é feita para auxílio do grupo que vai implementar a nossa especificação.

Neste para descrever a nossa forma de migração e a do outro grupo:

Se for por ficheiro temos de avaliar as várias opções, mas sem realizar testes penso que o mais vantajoso poderá ser um único ficheiro csv com diversas sheets correspondentes às várias tabelas. O custo que tens em abrir e fechar ficheiros é apenas um, e a informação fica na mesma facilmente organizada, a desvantagem é que não é seguro e o ficheiro podes ser consultado e alterado. Por sql também poderá ser uma boa opção, mas só vendo com testes de performance, segurança e estabilidade.

Por php teríamos de usar o código que o professor forneceu alterando para abrir as duas bds e copiar diretamente a informação de um lado para o outro

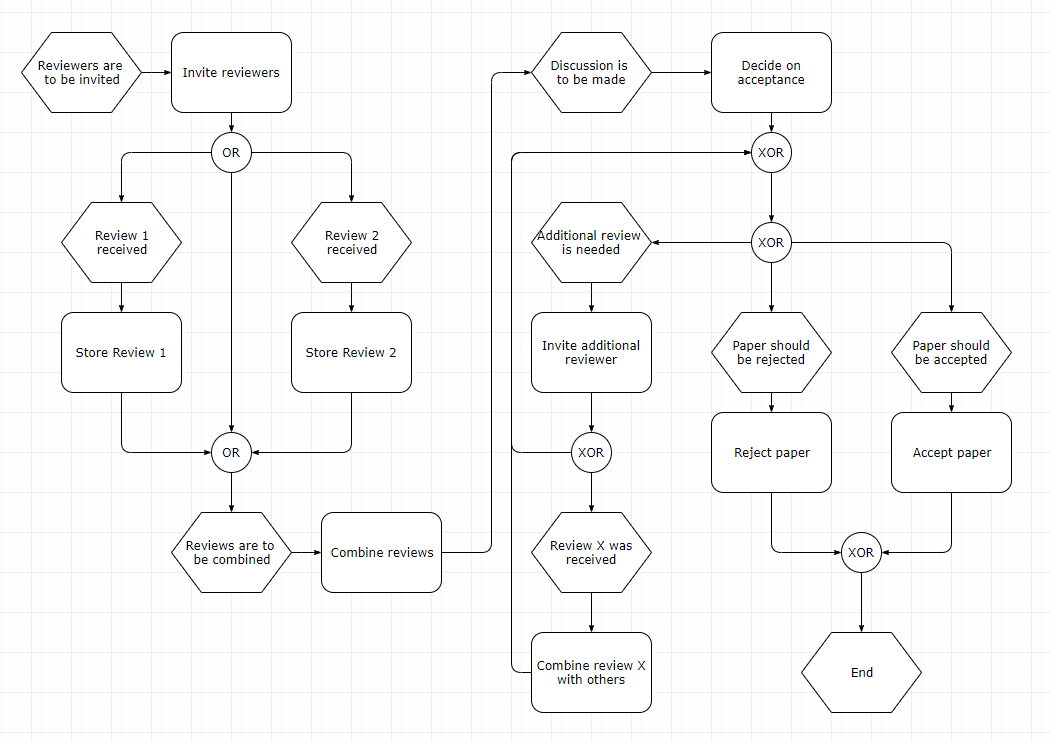
<Nesta secção deverá ser explicado a forma como a migração será efectuada. A informação apresentada deverá ser suficiente para que o grupo que a receba consiga implementar a migração.

Poderá ser importante incluir um diagrama que explique as várias etapas.

Deverão ser enumeradas as opções, comandos, utilitários, bibliotecas a utilizar. Caso relevante, indicar estratégias para nomes de ficheiros. Deverão ser abordadas questões como periocidade, privacidade dos dados, eficiência, facilidade de manutenção.

Quem especifica PHP não deve colocar o código, apenas a “lógica” subjacente.

Deverá estar claro no texto de que forma se assegura uma migração incremental (evitar que a informação chegue repetida à base de dados destino)>



Podemos usar um diagrama deste género para indicar os diversos passos a realizar para a migração.

Este ponto é um dos mais importantes do relatório, devemos indicar o porquê das nossas escolhas e além disso todas as outras opções que tivemos em conta e o porque de não termos escolhido as outras. Se fizermos desta forma temos 20 (dito pelo prof )

1.4.3 – Gestão de Utilizadores de Suporte à Migração (origem e/ou destino)

Auditor e tabelas a que ele acede e que apenas pode fazer select.

<Nesta secção deverá ser explicado de que forma deverá ser feita a manutenção (e.g., a criação e privilégios) de utilizadores (quer no destino, quer na origem, se necessário). Nomeadamente deverá ser indicado, para cada tipo de utilizador, que privilégios ele tem sobre que tabelas e Stored procedures (caso sejam referenciados na secção 1.4.5).

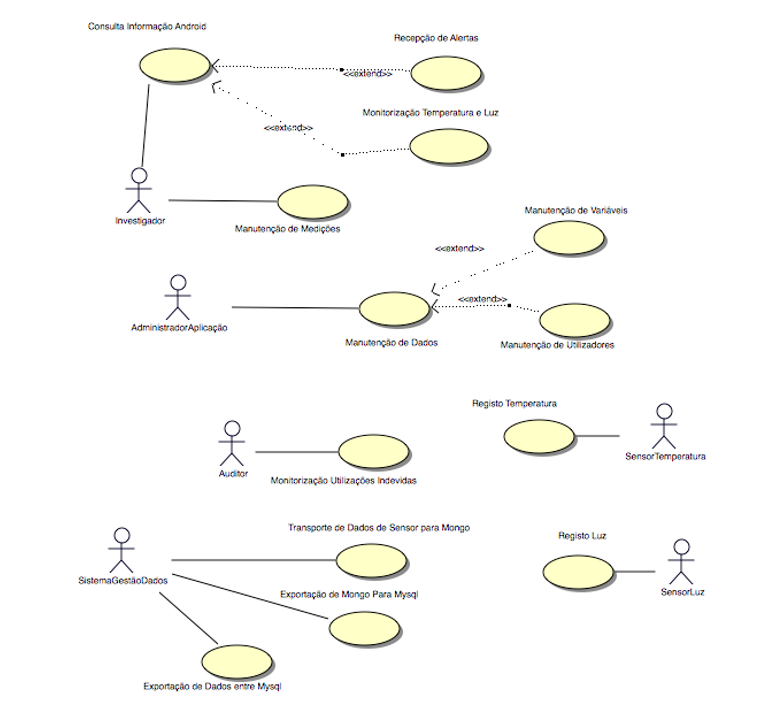
Sugestão de tabela

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Base de Dados (O/D)** | **Tabela** | **Tipo de Utilizador** | | |
| Tipo 1 | Tipo 2 | … |
|  | T1 | E | - |  |
|  | T2 | L | E |  |
|  | … |  |  |  |
|  | **Stored Proc.** |  |  |  |
|  | SP1 | X | - |  |
|  | … |  |  |  |

Em que E=Escrita, L=Leitura, X=Executar e - = sem permissões

Caso seja pertinente poderão ser adicionados comentários explicar a estratégia a seguir.>

Como se pode ver na tabela é também referido O/D, ou seja, Origem e Destino. Por isso penso que todos os utilizadores envolvidos no processo deverão ser considerados neste ponto. “deverá ser explicado de que forma deverá ser feita a manutenção (e.g., a criação e privilégios) de utilizadores (quer no destino, quer na origem, se necessário)”



Os restantes pontos do relatório acho que não se aplicam por que o auditor não vai usar triggers nem Stored Procedures. E o que se refere ao PHP ainda não sei como funciona.

1.4.4 - Triggers de suporte à migração de dados (origem e/ou destino) (se relevante)

Tal como se lê no título, Origem e destino significa que estes triggers não são apenas para o auditor, mas são triggers usados no processo de migração.

Por exemplo, se tivermos uma tabela de registo de migrações, para sabermos se as migrações ocorreram bem, data e hora, que tabela foi migrada, quantos registos foram copiados e se foi bem sucedida ou não, teremos de ter triggers para registar este tipo de informação nesta tabela automaticamente.

1.4.5 - Stored Procedures de suporte à migração de dados

Este ponto é muito importante, é através de SP que o Auditor acede à BD, este apenas deve conseguir realizar select e não pode inserir ou alterar.

Além disso temos o utilizador SistemaGestãoDados que tem de ter SP para inserir registos, no processo de migração.

1.4.6 Eventos de suporte à migração de dados

É neste ponto que iremos relatar eventos lançados em caso de erros ou avisos. Por exemplo: se ao migrar os dados o SO dá erro porque não existe mais espaço no disco e não consegue gravar mais nada, esse evento tem de ser relatado aqui. Isto é eventos que executam SP automaticamente tendo a possibilidade de por uma hora ou de x em x tempo. Aqui é útil para passar informação para o ficheiro.

1.4.7 PHP suporte à migração de dados (se relevante)

Terei de rever este tópico no momento em que se especificar o php.

Notas: Acho que existe alguma confusão com os termos *migração* e *destino*. De acordo com as imagens em cima temos de ter 2 BDs MySQL, uma de origem e outra de destino.

A de **origem** tem tabelas de log que registam os logs necessários além das tabelas normais.

A de **destino** é um “backup” dos logs e eventualmente também das medições. A consulta da BD de destino pelo auditor é realizada através da interface phpMyAdmin, no entanto a BD que se encontra por trás é MySQL.

O Processo de migração é a forma como vamos transferir a informação de uma Base de dados para a outra.

Mais tarde teremos também uma terceira BD a MongoBD que regista as informações dos sensores, que será transferida para as tabelas de medições por JDBC.