

Primeiros passos com o MySQL Workbench

FEB 16, 2014

Entenda rápido como começar com o MySQL Workbench

Hoje em dia, utilizo bastante o MySQL Workbench para manter a estrutura de meus BDs, ao invés do phpmyadmin. Eu prefiro, pois tenho um ambiente visual e facilidade em visualizar as alterações.

Para exemplificar, vou criar algo bem simples, mas que explora as vantagens que quero mostrar. Imagine que sua empresa possua franquias, e essas franquias possuam funcionários e clientes, e precisamos respeitar o seguinte:

- Ao desligar uma franquia, descadastramos junto seus funcionários;
- Ao desligar uma franquia, mantêmos seus clientes cadastrados.

Podemos resolver isso com a programação no server side, mas conseguimos resolver isso mais fácil pelo MySQL, e com mais segurança (pois já temos restrições bem testadas no BD).

Antes de começar

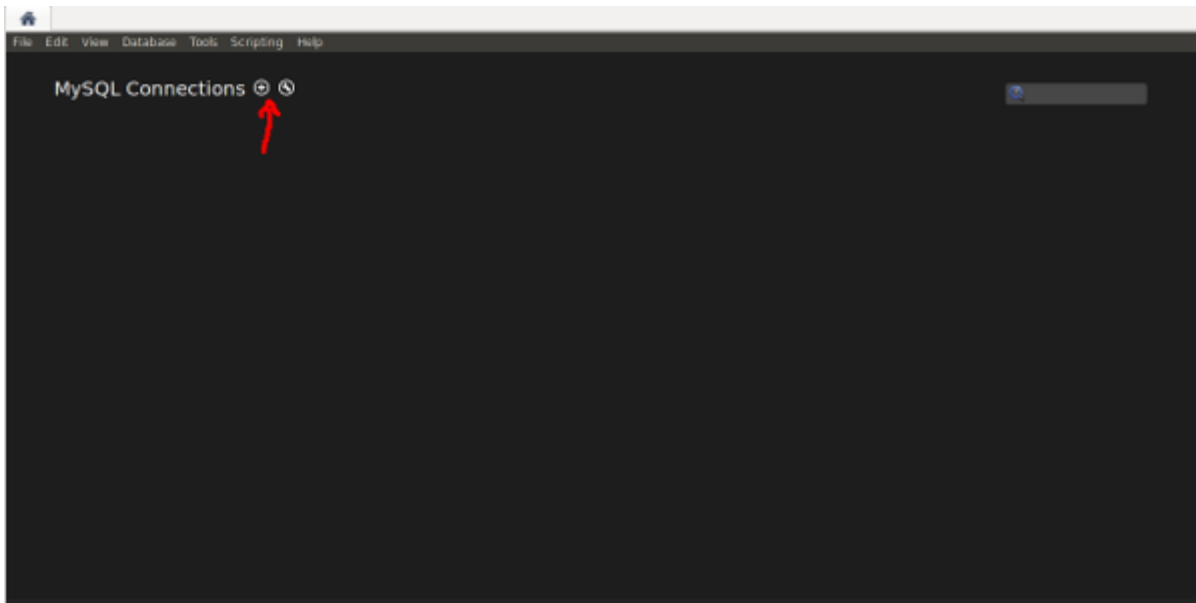
Tenha ele instalado: [Link para download](#)

Estou utilizando a versão 6.0.

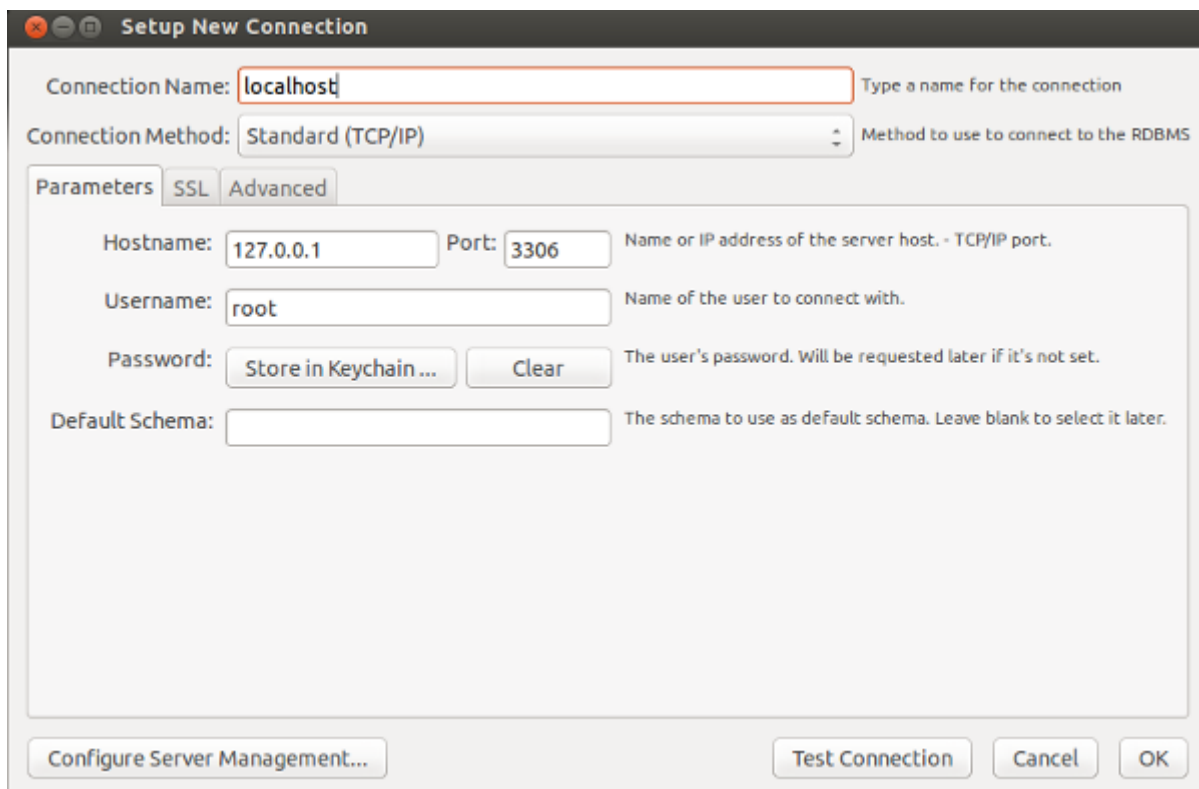
Iniciando um projeto

Vamos começar criando uma conexão com o Banco de Dados, e então criando um database de teste. (Obs.: Você pode pular esse passo, se já tiver criado um database para isso)

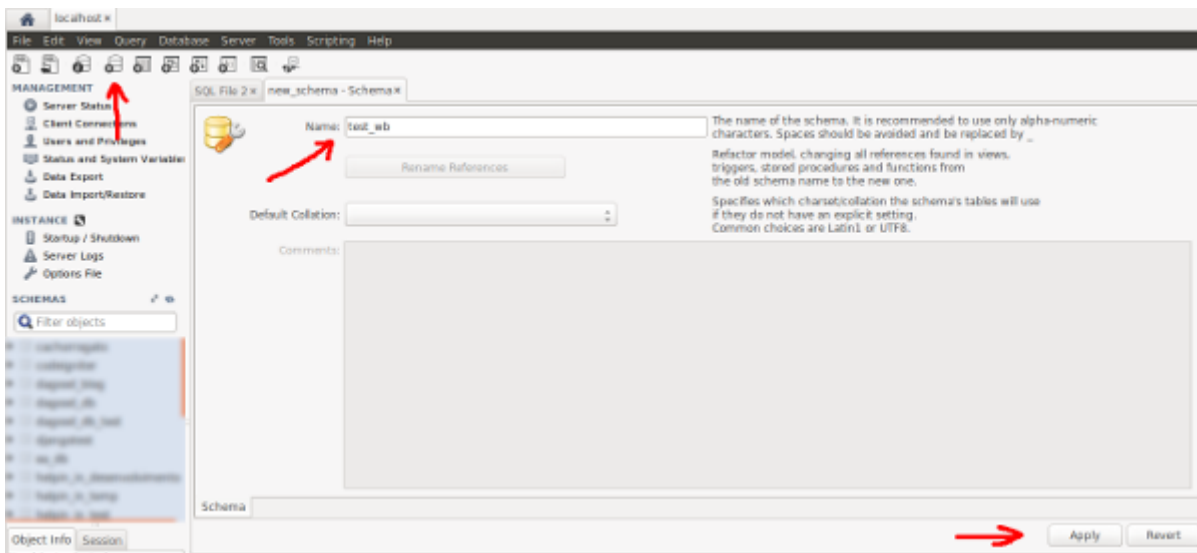
Clique no sinal de “+” no começo da janela.



Em seguida, ele vai perguntar as informações de conexão. Se for o padrão localhost, você vai precisar preencher apenas um nome para essa conexão (no meu caso, chamei de localhost mesmo). Ele deve pedir a senha também (se você ainda não tiver armazenado).



E por fim, vamos criar um novo database:

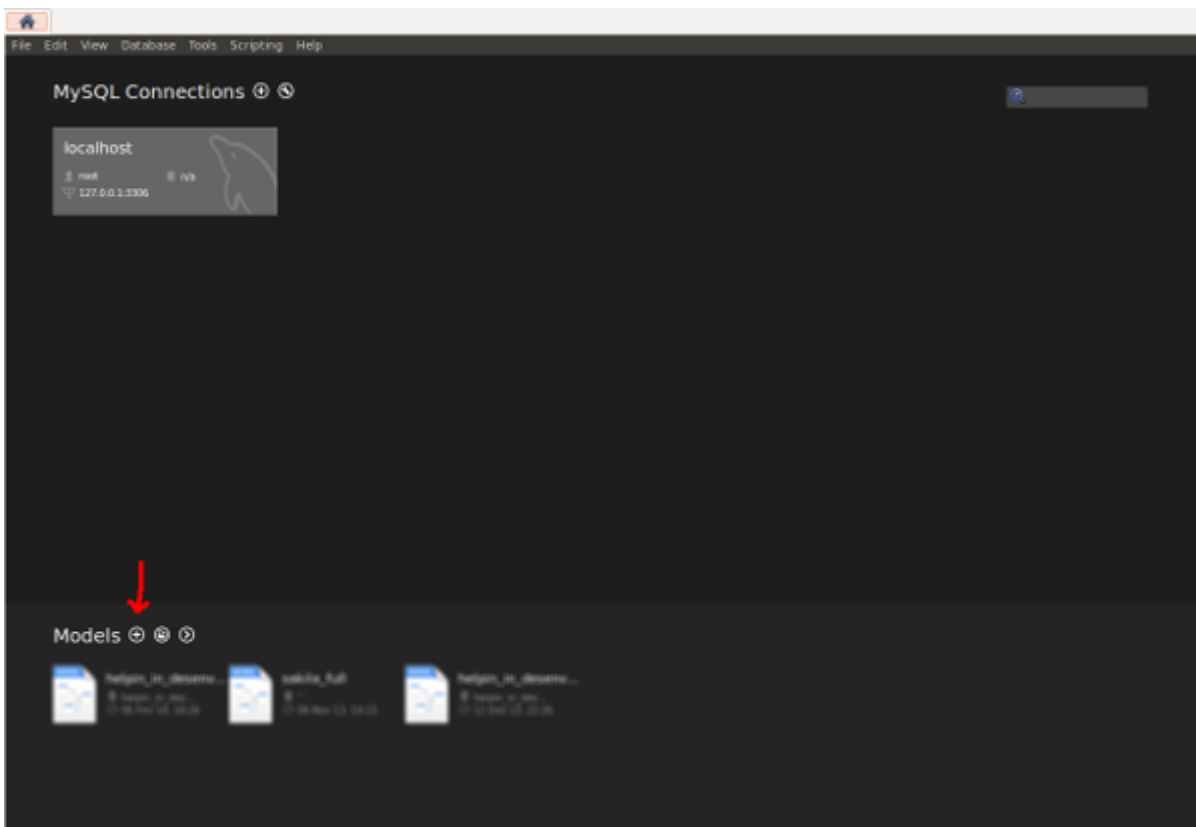


Clique no primeiro desenho (new schema), em seguida preencha o nome de seu database (chamei de test_wb) e por fim clique em apply.

Se você estiver com dúvida do motivo de estar usando schema, ao invés de database, eles são a mesma coisa. O MySQL não faz distinção entre eles (veja mais em <http://stackoverflow.com/questions/11618277/difference-between-schema-database-in-mysql>)

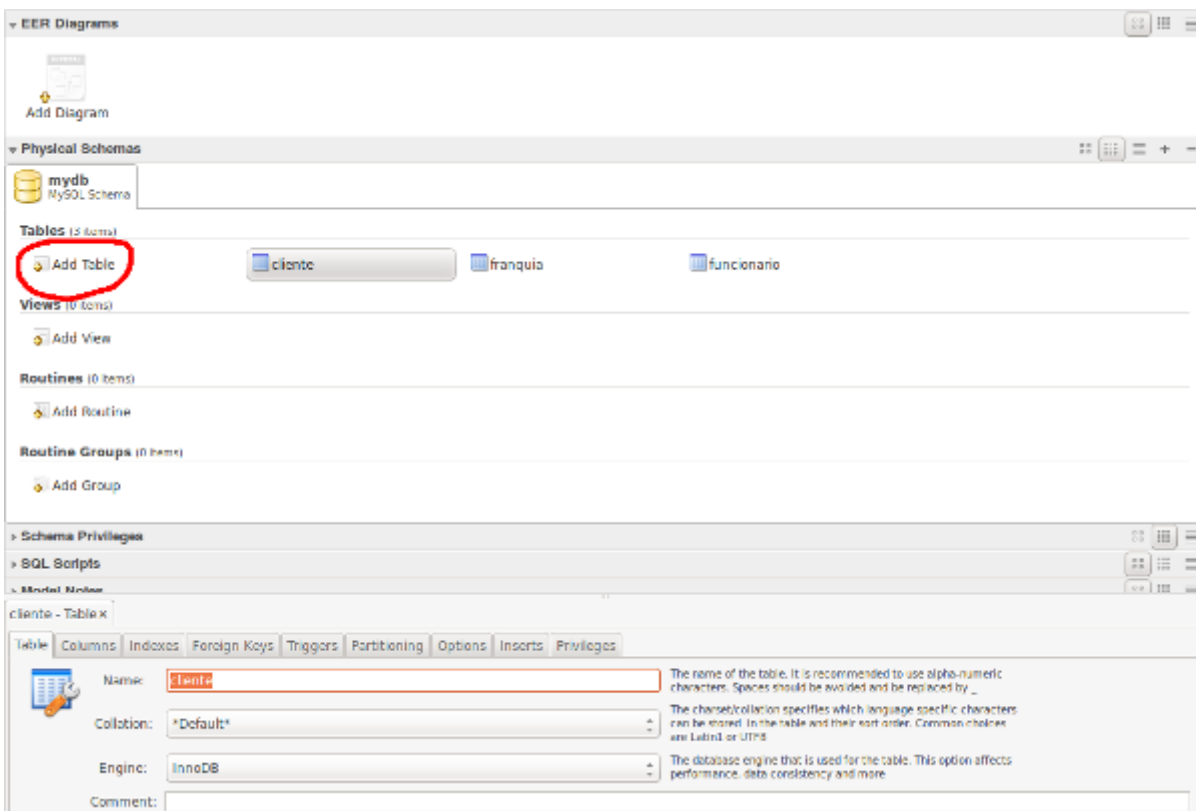
Criando um modelo

Feche a aba dessa conexão para voltar a página inicial. Clique no + para criar um novo modelo.



O programa vai abrir uma nova aba com um modelo sem nenhuma informação.

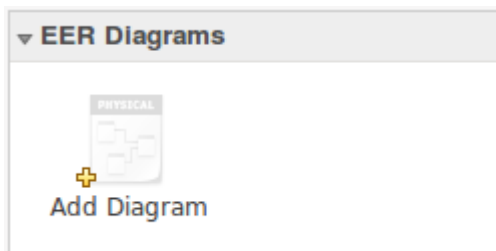
Para o nosso exercício, precisamos de 3 tabelas (franquia, funcionario, cliente). Vamos criá-las clicando em “Add table”, e então renomeando-as.



E então crie as colunas com as informações relevantes. Nesse exercício, vou apenas ter o id e nome para cada.

cliente - Table										
Table Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges										
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
idcliente	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
nome	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

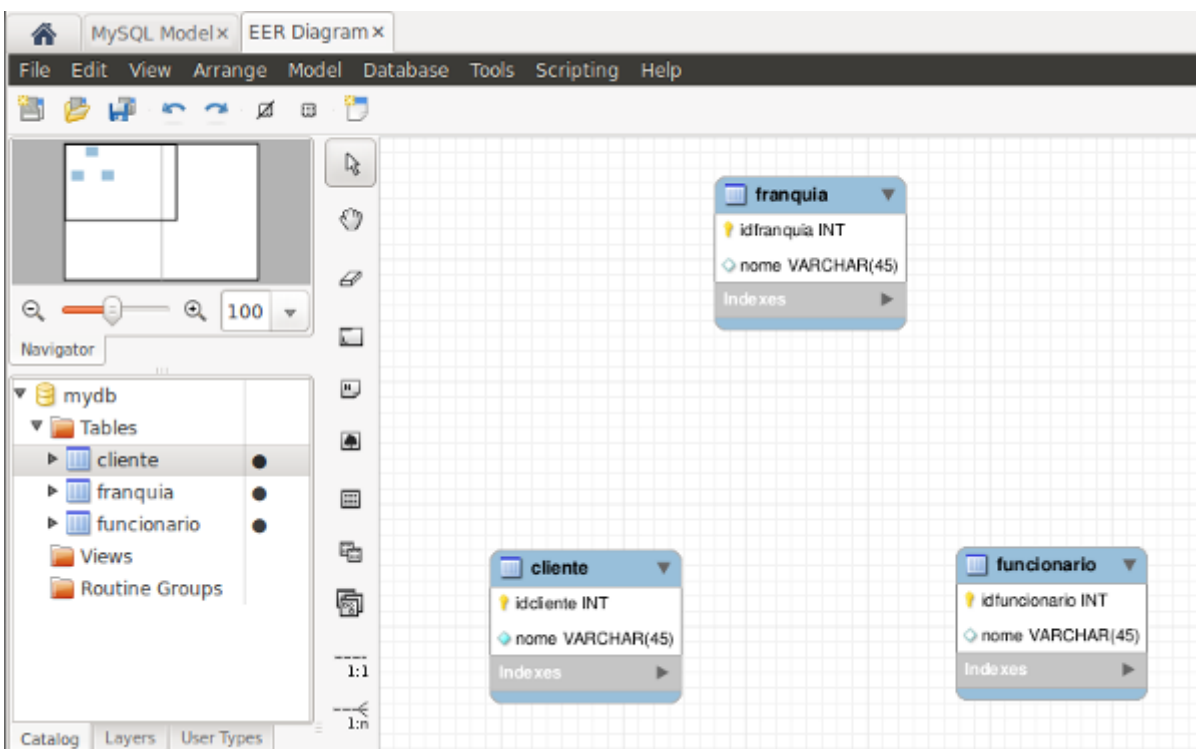
Crie um diagrama novo. Com ele, nós vamos visualizar graficamente os relacionamentos da tabela.



Trabalhando com o diagrama

Nesse momento, temos um diagrama vazio.

Usando o catálogo, e arrastando as tabelas que acabamos de criar, teremos o seguinte esquema (sem nenhum relacionamento).



À partir de agora, veremos a grande vantagem do MySQL workbench. Use as ferramentas para criar os relacionamentos.

Use a primeira 1:n, e então clique em cliente e depois em franquias. Com isso, você criou o primeiro relacionamento, em que cliente se relaciona com franquias através de uma coluna (no caso, chamada de “franquias_idfranquias”).

Agora use a segunda 1:n, e então clique em funcionário e depois em franquias. Agora, criamos outro relacionamento, porém esse identifica o funcionário como se fosse dessa franquias em específico.

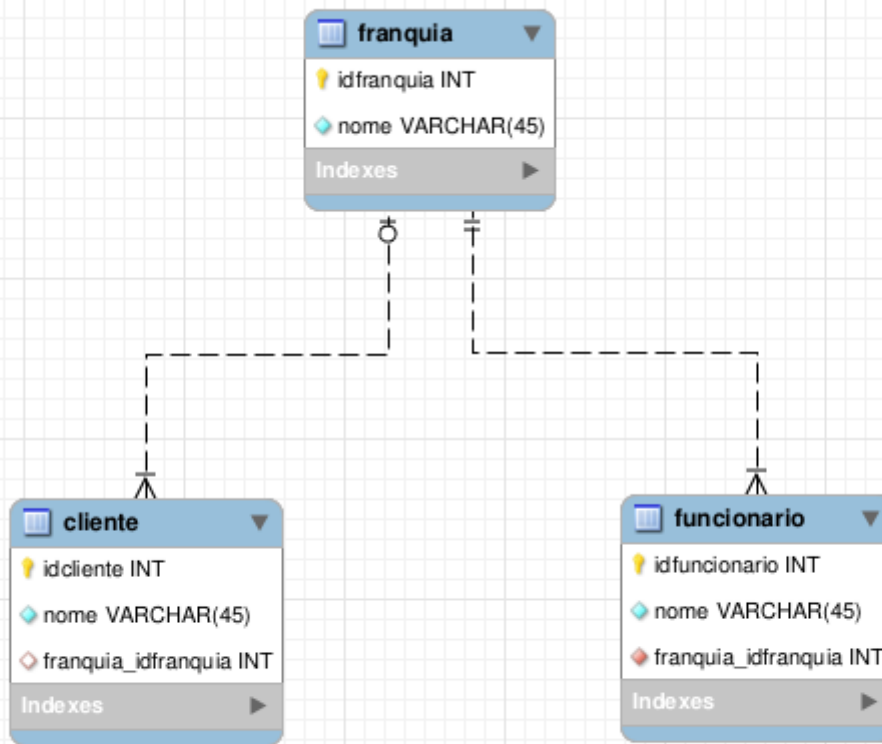
Além disso, eu prefiro fazer outras coisas. Edita as colunas da seguinte forma:

funcionario - Table x											
Table Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges											
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default		
idfuncionario	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>			
nome	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
franquias_idfranquias	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

cliente - Table x											
Table Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges											
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default		
idcliente	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>			
nome	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
franquias_idfranquias	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

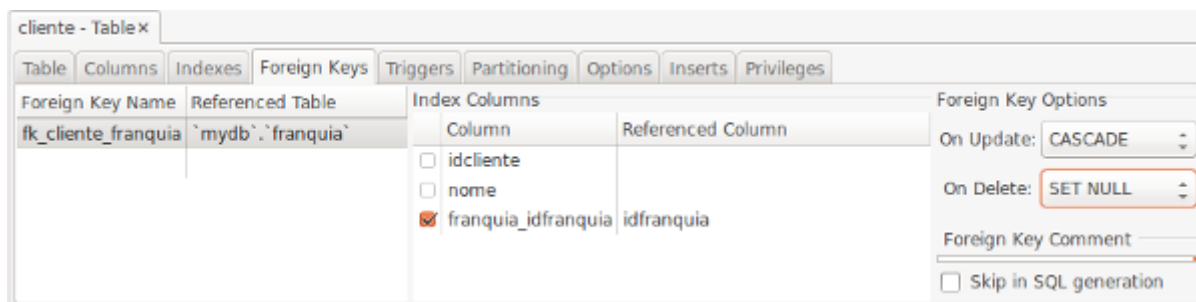
Eu fiz isso dessa forma, pois quero habilitar o NULL apenas para o cliente (pois tenho a intenção de haver cliente sem empresa), e não tenho a necessidade de que franquias_idfranquias seja identificador do funcionário.

Se estamos no mesmo passo, seu novo diagrama deve ser semelhante ao seguinte:



Configurando os relacionamentos

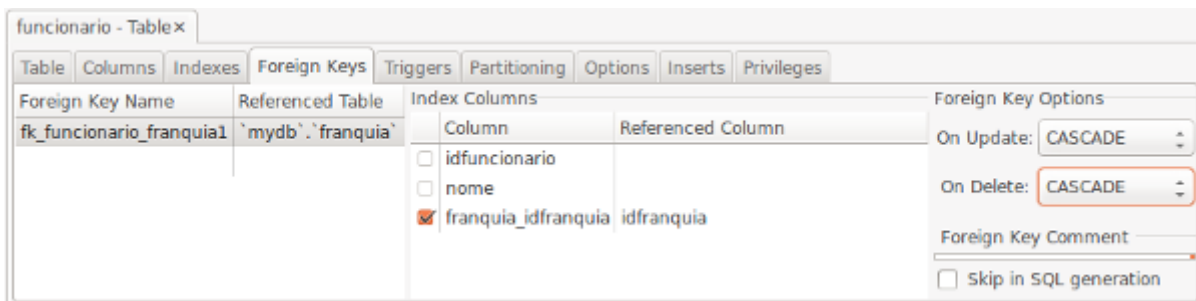
Estamos no final. Selecione a tabela de clientes, e a aba de Foreign Keys. Em Foreign Key Options, deixe CASCADE para Update e SET NULL para Delete.



Com isso, você diz para o BD que ao atualizarmos franquia, ele vai atualizar os clientes também. Porém, ao apagá-lo, você vai deixar os clientes com franquia desconhecida (franquia = NULL). É possível encontrá-los depois com uma query simples:

```
SELECT * FROM `cliente` WHERE franquia_idfranquia IS NULL
```

E por fim, configurar funcionario com CASCADE para ambos.



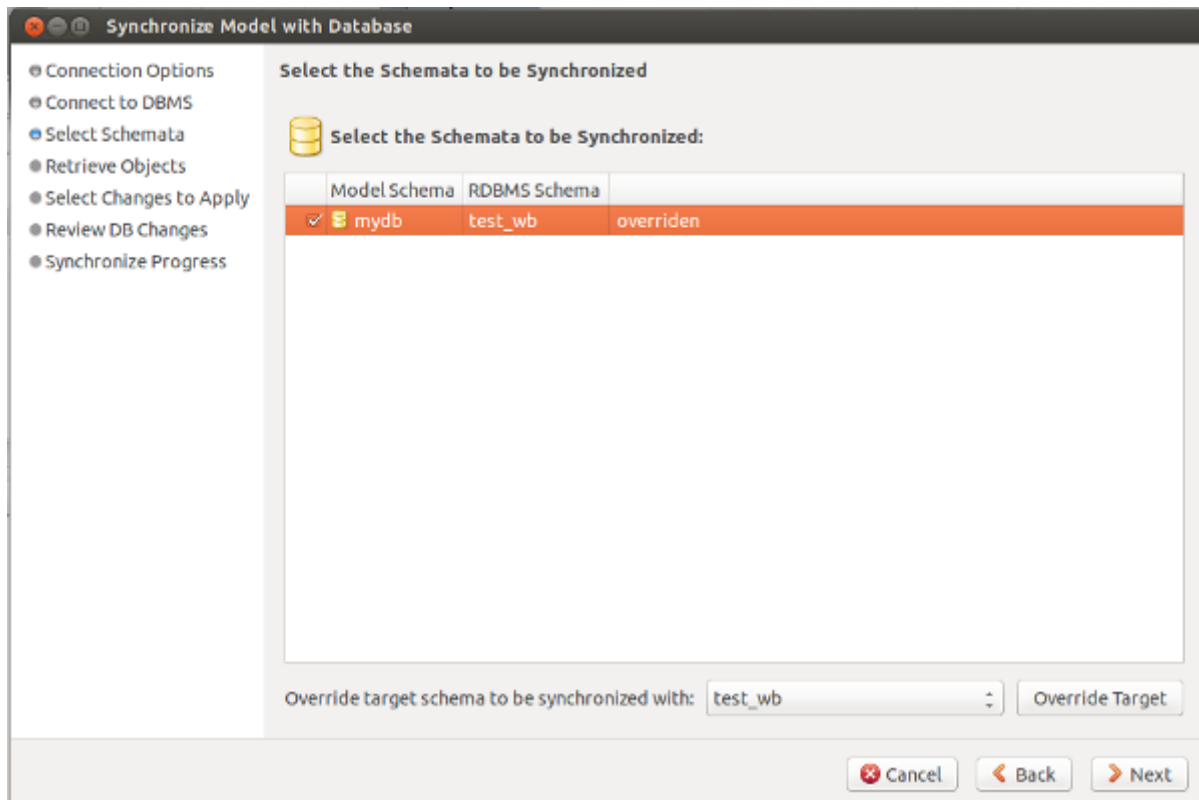
Com isso, ao atualizar ou apagar a franquia, você fará o mesmo com seus funcionários.

Deploy

Está pronto. Precisamos apenas colocar isso no BD.

Salve (ctrl + S). Exporte (Database -> Synchronize Model ou então ctrl + shift + Z).

Nessa nova janela, clique em Next, depois Next novamente. E então escolha o database (estou utilizando test_db, que criei no começo do tutorial), e clique em Override Target.



Por fim, clique em todos os Next e Execute. Está pronto. Você configurou corretamente seu BD para necessidade do exercício.

Como funciona

Agora, vou inserir alguns exemplos:

Inserir informação

```
INSERT INTO `test_wb`.`franquia` (`idfranquia`, `nome`) VALUES (NULL, 'franquia');  
INSERT INTO `test_wb`.`cliente` (`idcliente`, `nome`, `franquia_idfranquia`) VALUES  
INSERT INTO `test_wb`.`funcionario` (`idfuncionario`, `nome`, `franquia_idfranquia`
```

Atualizar franquias

```
UPDATE `test_wb`.`franquia` SET `idfranquia` = '2' WHERE `franquia`.`idfranquia`  
Após essa operação, é atualizado cliente e funcionário com id de franquias=2.
```

Apagar franquias

```
DELETE FROM `test_wb`.`franquia` WHERE `franquia`.`idfranquia` = 4  
Após essa operação, todos os funcionários da franquias são apagados, mas mantemos os
```

Cuidado

Na primeira vez que mexi com o MySQL Workbench, errei bastante em definir uma ação em On Delete como SET NULL, mas não deixei o campo permitir NULL. Com isso, dava erro ao sincronizar ambos, e não informava qual o erro. Levei um bom tempo para descobrir onde era a falha.

Conclusão

Hoje em dia, utilizo bastante o MySQL em meus projetos PHP para criar e manter esses relacionamentos. Acho muito mais fácil do que escrever os códigos no phpmyadmin. O phpmyadmin eu utilizo para pequenas operações (como visualizar, editar e apagar dados), mas isso também é possível fazer com o Workbench.