



ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO  
INSTITUTO POLITÉCNICO DE LEIRIA

## ***Sistemas de Bases de Dados***

ENGENHARIA INFORMÁTICA 2019/20

2.º ano, 2.º semestre

### **Ficha de Trabalho n.º 1 – Segurança em Bases de Dados (parte I)**

#### **Objetivo:**

- Gerir utilizadores e seus privilégios no acesso a dados

Antes de iniciar a resolução desta ficha de trabalho deverá responder às seguintes questões:

**1.** Diga o entende por:

- Utilizador;
- Privilégio de sistema;
- Privilégio de objeto;
- Role*;

**2.** Apresente um exemplo do comando que permite realizar cada uma das seguintes ações:

- Criar um utilizador; `CREATE USER <nome> IDENTIFIED BY '<password>'`
- Conceder um privilégio de sistema a um utilizador; `GRANT CREATE SESSION TO <nome>`  
`CREATE ROLE <nome>`
- Conceder um privilégio de objeto a um *role*; `GRANT SELECT ON <tabela> TO <nome.role>`
- Associar um *role* a um utilizador; `GRANT <nome.role> TO <nome.user>`
- Retirar a um *role* um privilégio de sistema que lhe haja sido concedido.  
`REVOKE <privilegio> FROM <nome.user>`

## CASO DE ESTUDO

Uma determinada escola de condução dedica-se ao ensino da condução a membros da comunidade com mais de 18 anos de idade. Para se ser aluno e ter acesso às aulas de condução é necessário realizar uma inscrição. Para obter aprovação à categoria automóvel de uma inscrição, cada aluno deve realizar um exame (se reprovar no exame, terá de fazer nova inscrição). Cada exame é preparado pela Direção Geral de Viação para vários alunos: cada aluno obtém a categoria respetiva assim que o resultado do exame for definido, sendo a data do exame a que define a data de obtenção da categoria correspondente. Quando uma inscrição for paga, a data de pagamento será registada e o valor do atributo *paga* será automaticamente atualizado.

Considere que a base de dados utilizada para armazenar toda a informação contém as tabelas da Figura 1, onde as **chaves primárias estão a negrito e sublinhadas** e as **chaves estrangeiras estão a negrito e itálico**:

<b><u>bi</u></b>	nome	morada	data_nasc	última_categoria_obtida	data_última_categoria_obtida	total_reprovacoes
10700007	Carlos Sousa	Rua das Tijoleiras	26-02-1997	NULL	NULL	2
10800008	Susana Costa	Rua da Beleza	29-08-1984	NULL	NULL	1
10900009	Filipe da Silva	Av. Vidal Pinheiro	01-01-1995	C	05-02-2019	0

### INSCRICAO

<b><u>cod_inscricao</u></b>	data_insc	paga	data_pagamento	categoria	<b><u>bi_aluno</u></b>	<b><u>id_exame</u></b>	resultado_exame
7089	08-11-2019	S	08-12-2019	C	10900009	10901	Aprovado
7090	01-12-2019	S	04-01-2020	A	10700007	10900	Reprovado
7091	08-12-2019	S	04-01-2020	C	10800008	10901	Reprovado
7092	10-01-2020	N	NULL	B	11100000	NULL	NULL
7093	11-01-2020	S	12-01-2020	A	10700007	10902	Reprovado
7094	10-01-2020	S	20-01-2020	A	10700007	10903	NULL
7095	08-02-2020	N	NULL	D	10700007	NULL	NULL

### EXAME

<b><u>id</u></b>	local	data	categoria
10809	Estádio da cidade	03-12-2009	C
10900	Estádio da cidade	04-12-2019	A
10901	Escola	01-01-2020	C
10902	Estádio da cidade	07-02-2020	A
10903	Escola	09-02-2020	A
10904	Centro de Testes Automóveis	12-03-2020	B
10905	Centro de Testes Automóveis	13-03-2020	A
10906	Centro de Testes Automóveis	01-04-2020	B
10907	Estádio da cidade	10-04-2020	B

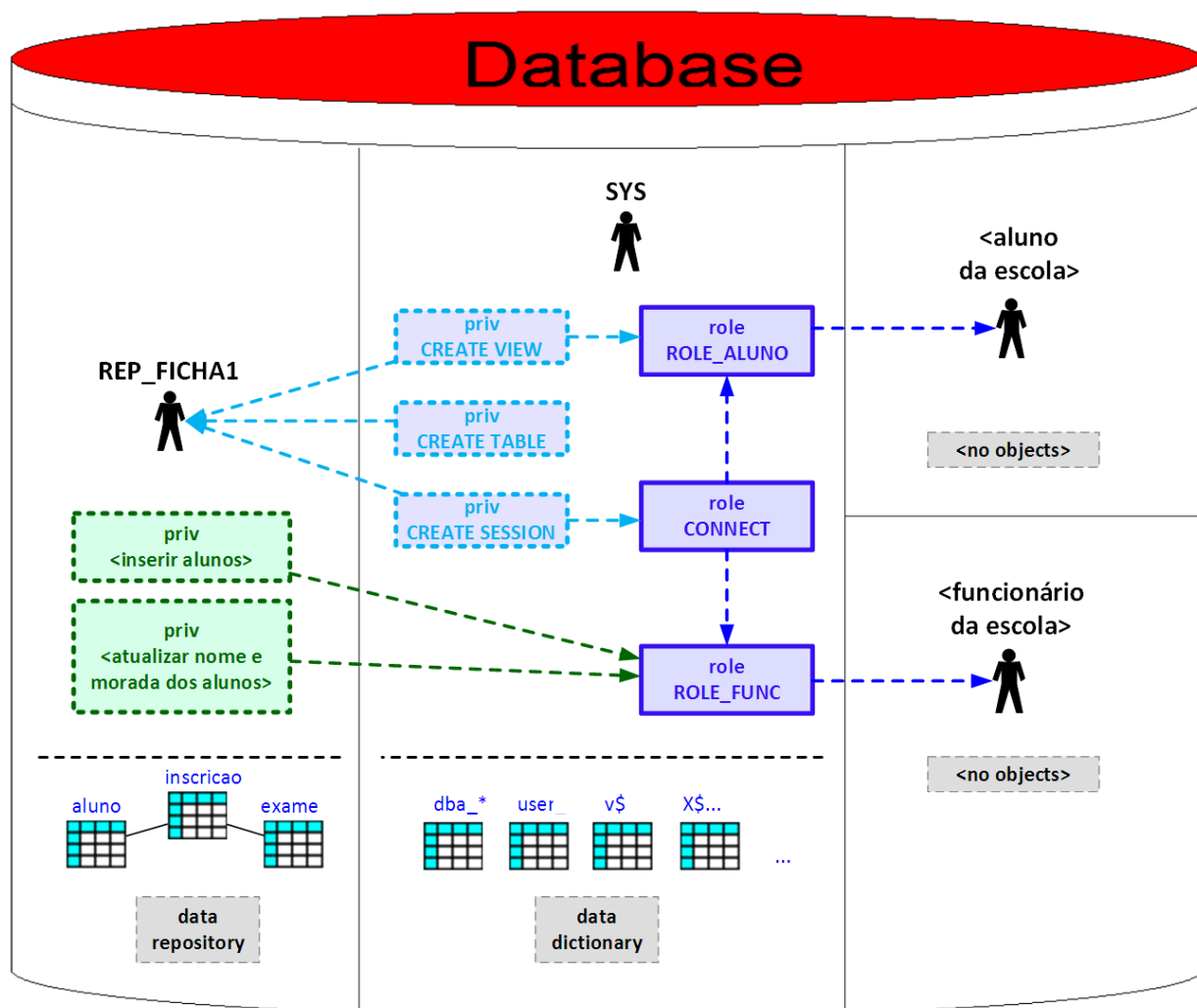


Figura 2 – Arquitetura **pretendida** para os utilizadores da base de dados.

## Notas prévias

Nesta ficha de trabalho, assumirá tanto o papel de DBA (*Database Administrator*, utilizador SYS) como de programador de base de dados (utilizador REP\_FICHA1). Assuma esses papéis levando a cabo boas práticas de administração e de programação:

- Analise todo o cenário de antemão e tenha-o sempre **rapidamente acessível**, pois é importante ter-se em todos os instantes a visão global do sistema que se administra ou no qual se programa.
- Seja **eficaz e eficiente** nas soluções aplicadas: os recursos são sempre limitados e as soluções aplicadas deverão ter sempre com essa noção em mente. Não distribua mais recursos ou privilégios do que os estritamente necessários.
- Antes de aplicar as suas soluções, **pense no impacto** que a implementação das mesmas terá no sistema.
- Verifique e teste** todas as alterações realizadas sobre os objetos ou privilégios da base de dados, consultando os objetos envolvidos e/ou o dicionário de dados.
- Guarde num ficheiro sql** a sequência exata e completa dos comandos executados, juntamente com os apontamentos relevantes sobre o contexto da execução. Este ficheiro, quando executado de forma integral, deverá permitir resolver na íntegra toda a ficha.

Após terminar os exercícios, renomeie o ficheiro para <n.º estudante\_SBDficha1.1 \_parte1.sql> e submeta-o no Moodle utilizando o link apropriado (por exemplo, o estudante n.º 21000001 submeterá o ficheiro 21000001\_SBDficha1.1 \_parte1.sql).

Os exercícios assinalados com (\*) deverão ser realizados em estudo autónomo.

1. Instale o cenário no utilizador REP\_FICHA1. **No final de cada alínea utilize o dicionário de dados para verificar as alterações ocorridas na base de dados.**  
login: sys as sysdba / sys
  - a) Execute o *script* FICHA01\_USERS.SQL na conta apropriada.
  - b) Execute o *script* FICHA01\_TABELAS.SQL na conta apropriada.  
**Nota:** deverão ocorrer vários erros na criação das tabelas do cenário: procure compreender o porquê dos erros ocorridos e corrija eficientemente as situações incorretas antes de nova tentativa.
  - c) Execute o *script* FICHA01\_DADOS.SQL. Visualize a estrutura e os dados de cada tabela.
2. Implemente consultas para:
  - a) (\*) Apresentar o código e a data das inscrições com exames a realizar no ano atual considerando apenas uma determinada categoria a ser pedida ao utilizador.
  - b) (\*) Apresentar o nome dos alunos e a data em que cada um obteve aprovação na categoria C.
  - c) (\*) Apresentar o *bi*, o nome do aluno e a idade que cada aluno tinha na data em que obteve aprovação na última categoria.
  - d) (\*) Apresentar para cada categoria, quantos dias passam (em média) desde a data da inscrição até à data do pagamento da inscrição. A consulta deve mostrar primeiro as categorias com valor médio mais alto.
  - e) (\*) Apresentar o *bi* e o nome dos alunos com exame marcado para a categoria A e que já reprovaram anteriormente mais que uma vez nesta categoria.
3. Cada aluno da escola de condução deverá ter um utilizador atribuído na base de dados. O utilizador na base de dados deverá ter um *username* formado pelo primeiro nome do aluno seguido do *número de bi* desse aluno (a palavra passe deverá ser igual ao *username*). *Por exemplo, ao aluno Carlos Sousa deverá na base de dados ser atribuído o username CARLOS10700007 com a palavra passe CARLOS10700007.*
  - a) Altere a estrutura da tabela ALUNO de forma a que seja possível associar os dados de cada aluno ao utilizador que lhes corresponde na base de dados.
  - b) Averigue, a partir do dicionário de dados, quais os alunos da escola de condução que já possuem um utilizador associado na base de dados. Para estes alunos, atualize de forma **o mais automatizada possível** os dados da tabela ALUNO.
  - c) Para os alunos da escola de condução não abrangidos pela alínea anterior, crie o respetivo utilizador da base de dados e atualize os dados da tabela ALUNO.
4. Analise que privilégios possuem os alunos da escola de condução.
5. Com a ajuda da Figura 2, atualize os privilégios de cada aluno da escola de condução.
6. Um novo funcionário da escola de condução, o Sr. João Almeida, passará a ter acesso à base de dados de acordo com o que está definido Figura 2, através do utilizador JALMEIDA: garanta que tal é possível.  
(\*) Garanta que todos os funcionários podem consultar, atualizar e inserir inscrições e exames;
7. Implemente o mecanismo que permitirá a cada aluno da escola de condução visualizar o local, a data e a categoria dos exames agendados para os próximos 30 dias.
8. Implemente o mecanismo que permitirá a cada aluno da escola de condução visualizar unicamente os seus dados (*bi*, nome, morada e data de nascimento) e nunca dados de outros alunos.

9. (\*) Implemente o mecanismo que permitirá a cada aluno visualizar a data da inscrição, a data de pagamento, a data de exame, a categoria, o local e o resultado que obteve em cada um dos exames que já realizou.
10. (\*) Crie uma sequência que possa ser usada na geração de valores automáticos para a chave primária da tabela INSCRICAO e implemente o(s) *trigger(s)* que dela tirem partido.
11. (\*) Crie a função FUNC\_CALC\_ANOS que calcule o n.º de anos decorridos entre uma data e a data atual. Com esta função, imponha a restrição de integridade relativa à idade mínima dos alunos da escola.
12. (\*) Crie na base de dados os objetos necessários para garantir a integridade dos atributos desnormalizados *última\_categoria\_obtida* e *data\_última\_categoria\_obtida* sempre que for atualizado o campo *resultado\_exame*. **Garanta que os funcionários conseguem atualizar esta coluna.**
13. (\*) Crie o procedimento PROC\_INSERTIR\_ALUNO que, quando executado, registará um dado aluno e inscrevê-lo-á numa determinada categoria.  
Recorde que a inserção de alunos no repositório de dados está a cargo dos funcionários.