

Ciência de Dados

Licenciatura Engenharia Informática
2º Semestre – 2021/2022

Ricardo Jesus Ferreira
ricardojesus.ferreira@my.istec.pt

Base Dados Relacional

Processo de Criação BD Relacional

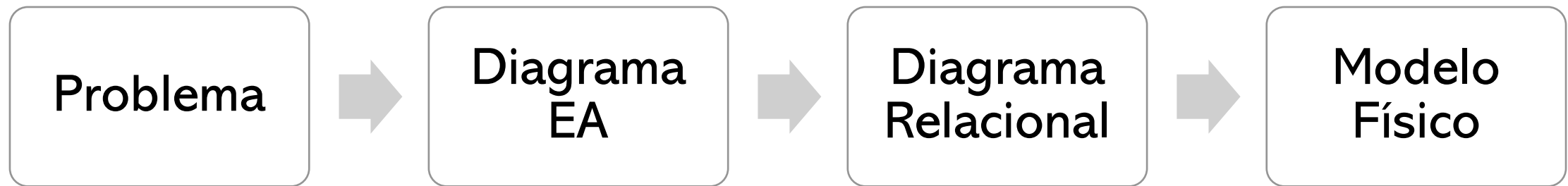
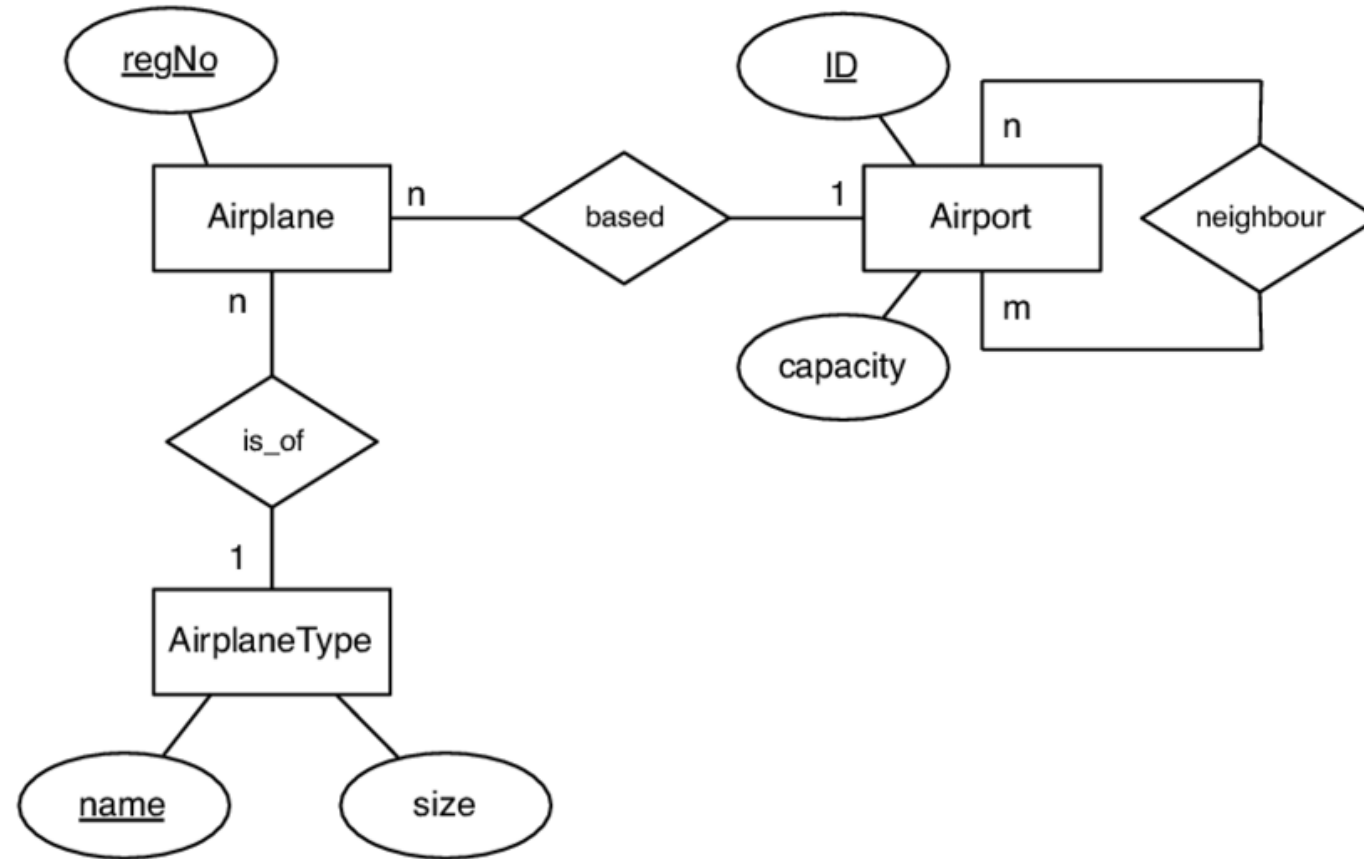


Diagrama Entidade-Associação

Diagrama Entidade-Associação

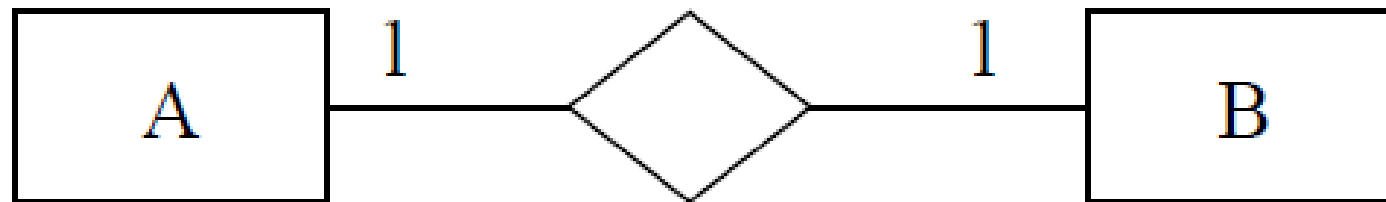
- Entidade
 - Qualquer coisa objeto ou conceito) com interesse para a organização , identificável inequivocamente , sobre a qual é guardada informação (e.g. Funcionário, Departamento, Contrato...)
- Atributo
 - Propriedades das entidades (Elemento atómico indivisível) de informação (e.g. N.º de empregado, Nome, ...)
 - Identificadores chaves
 - Descritores
- Associação
 - Relaciona entidades entre si (binária, complexa, unária)

Exemplo



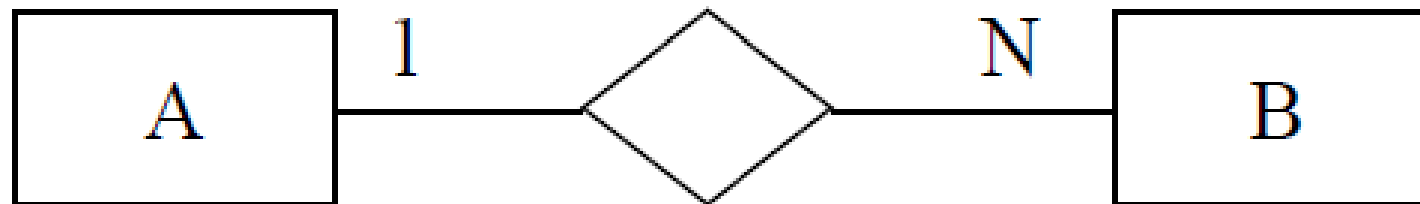
Associações

- Associação 1:1 (um para um)
 - A cada ocorrência da entidade A está associada no máximo uma ocorrência da entidade B (ou nenhuma)
 - A cada ocorrência da entidade B está associada no máximo uma ocorrência da entidade A (ou nenhuma)



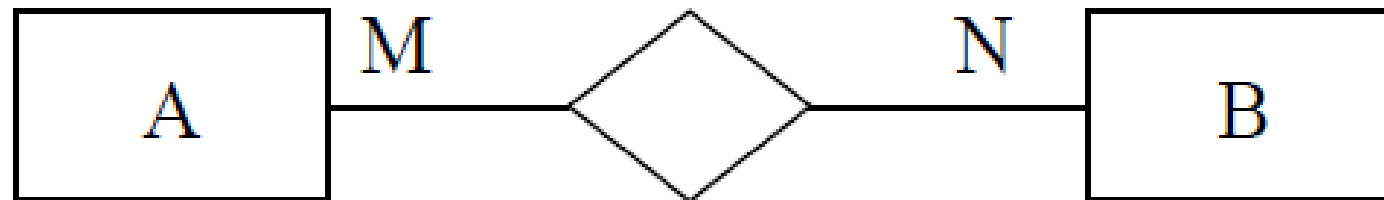
Associações

- Associação 1:N (um para vários)
 - A cada ocorrência da entidade A está associada uma, várias ou nenhuma ocorrência da entidade B
 - A cada ocorrência da entidade B está associada apenas uma ocorrência da entidade A (ou nenhuma)



Associações

- Associação M:N (vários para vários)
 - A cada ocorrência da entidade A está associada uma, várias ou nenhuma , ocorrência da entidade B
 - A cada ocorrência da entidade B está associada uma, várias ou nenhuma , ocorrência da entidade A



Erros comuns

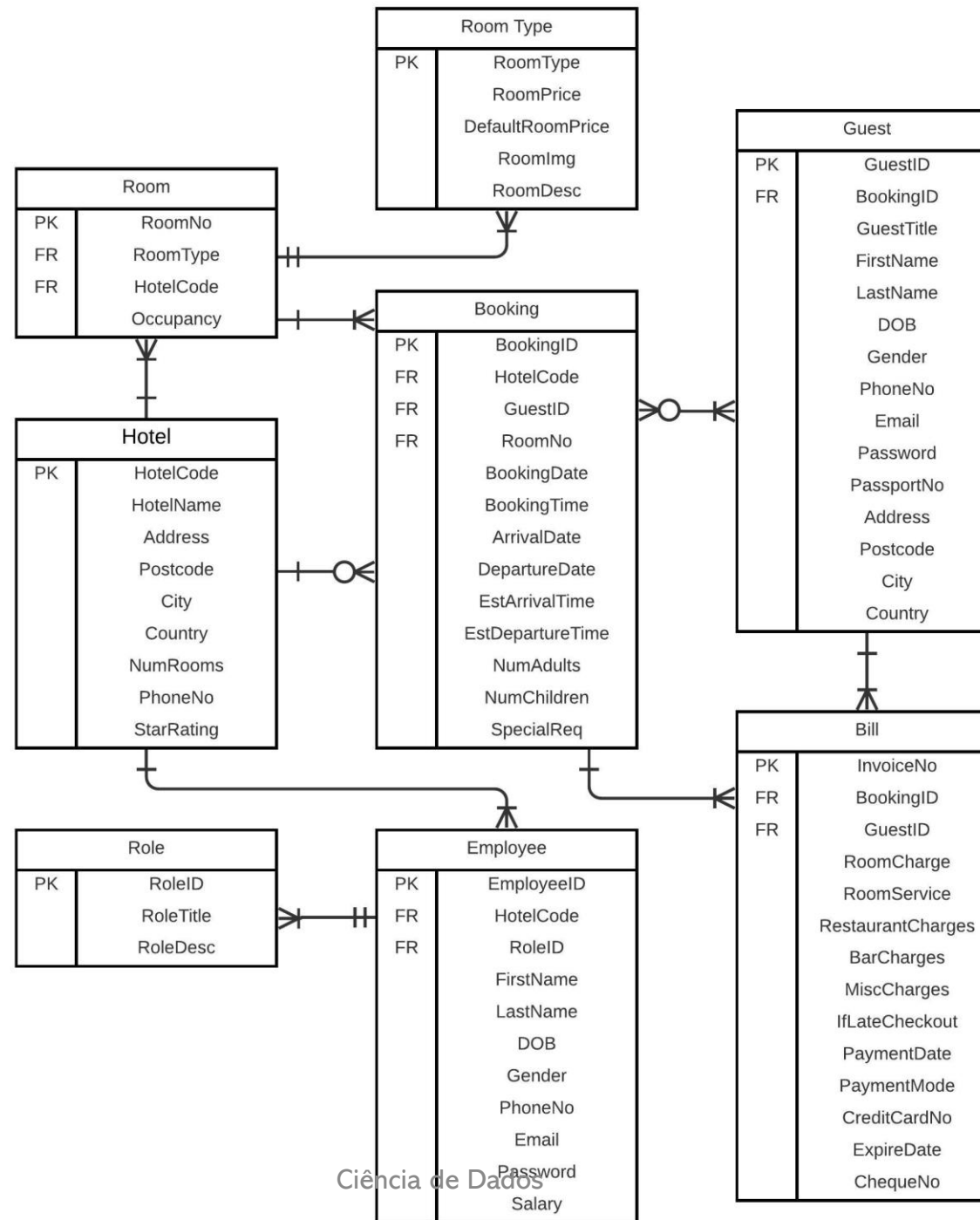
- Usar atributos que podem ser calculados (# funcionários)
- Entidades sem atributos
- Restrições anexas que podem ser modeladas
- Entidades com uma única instância (e.g. empresa para base de dados da empresa)
- Partes isoladas do diagrama
- Agregação de várias associações (agregação deve apenas conter uma associação)

Diagrama Relacional

Diagrama Relacional

- O modelo relacional é independente da implementação
- Define a estrutura de armazenamento de dados
- Um modelo formal simples e elegante baseado na teoria dos conjuntos: permite rápida deteção de anomalias

Exemplo



Chave Primária

- Conjuntos de um ou mais campos, cujos valores, nunca se repetem na mesma tabela e, desta forma, podem ser usadas como um índice de referência para criar relacionamentos com as demais tabela da base de dados.
- Portanto, uma chave primária nunca pode ter valor nulo, nem estar repetida.

Chave Estrangeira

- Uma chave estrangeira é um campo, que aponta para a chave primária de outra tabela ou da mesma tabela.
- A finalidade da chave estrangeira é garantir a integridade dos dados referenciais, pois apenas serão permitidos valores que supostamente vão aparecer na base de dados.
- Esse tipo de atributo não permite exclusão, modificação ou inserção de dados em tabelas que estejam dependentes umas das outras("foreign key").

Diagrama Relacional

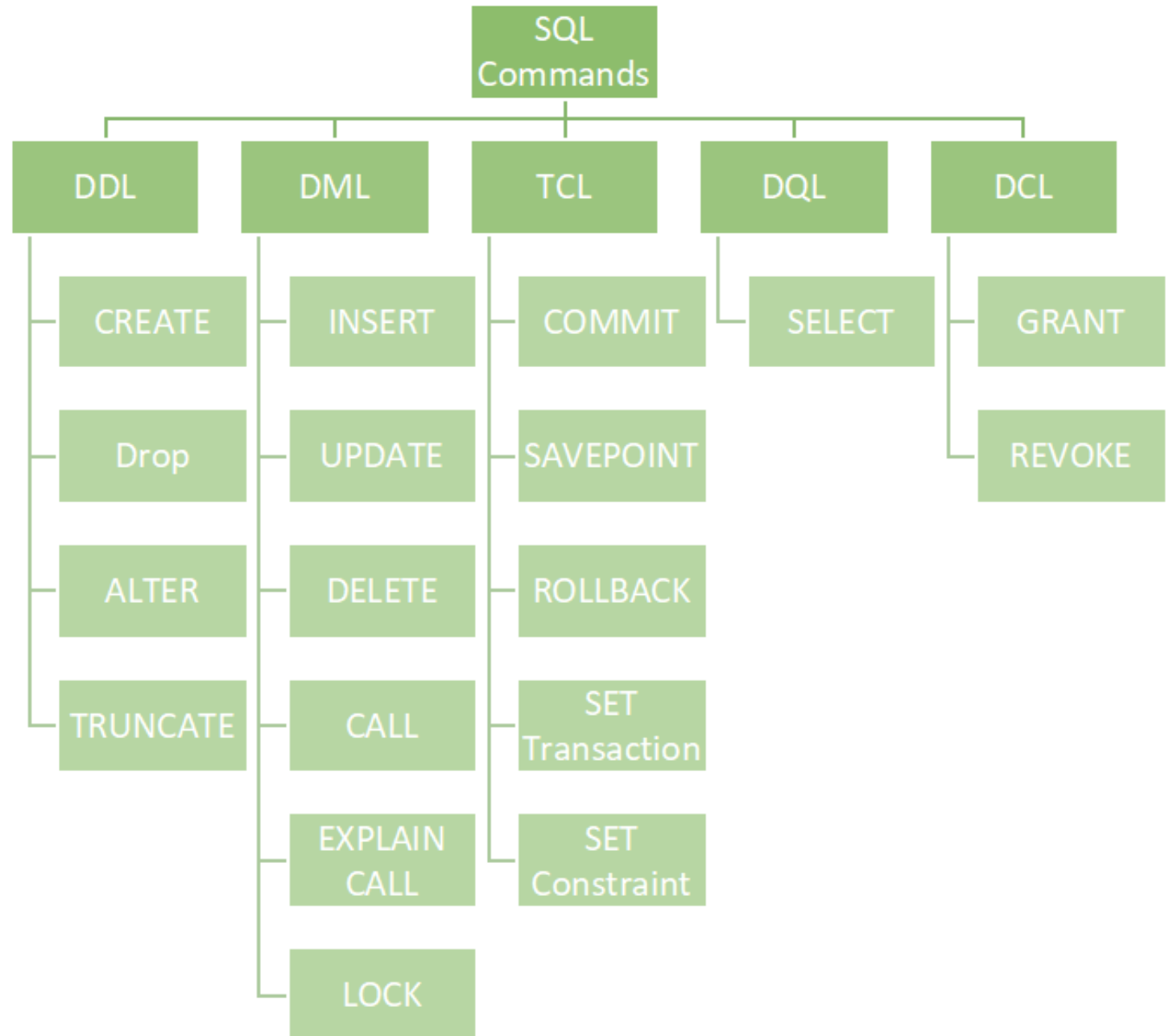
- Deve representar fielmente a realidade
- Seja capaz de responder às funcionalidades que se pretende
- Pretende se modelos que garantam
 - Redundância mínima
 - Facilidade de manutenção
 - Estabilidade face a futuras alterações

Modelo Físico

Modelo Físico

- No contexto do SQL, a definição de dados ou a linguagem de descrição de dados (DDL) é uma sintaxe para criar e modificar objetos de base de dados, tais como tabelas, índices e utilizadores.
- As declarações de DDL são semelhantes a uma linguagem de programação e são usadas para definir estruturas de dados, especialmente esquemas de base de dados.
- Exemplos comuns de declarações de DDL incluem CREATE, ALTER e DROP.

Categorias SQL



DDL - Data Definition Language

- Criar uma base de dados
 - CREATE DATABASE <nome bd>
- Selecionar uma base de dados para utilização:
 - USE <nome bd>
- Apagar uma base de dados:
 - DROP DATABASE <nome bd>

DDL - Data Definition Language

- Criar uma tabela
 - `CREATE TABLE <nome tabela> (<nome atributo> domínio, <nome atributo2> domínio2, restricoes_integridade);`
- Apagar uma tabela
 - `DROP TABLE <nome tabela>;`

DML - Data Manipulation Language

- Inserir valores numa tabela
 - INSERT INTO <nome tabela> VALUES <valor atributo1>, <valor atributo2>, <valor atributoN>);
- Alterar dados numa tabela
 - UPDATE <nome tabela> SET <nome atributo1> = <valor atributo1>, <nome atributo2> = <valor atributo2>, <nome atributoN > = <valor atributoN> WHERE <condição>;
- Apagar dados numa tabela
 - DELETE FROM <nome tabela> WHERE <condição>;

TCL - Transaction Control Language

- As transações agrupam um conjunto de tarefas numa única unidade de execução.
- Cada transação começa com uma tarefa específica e termina quando todas as tarefas do grupo terminam com sucesso.
- Se alguma das tarefas falhar, a transação falha. Portanto, uma transação tem apenas dois resultados: sucesso ou insucesso.

TCL - Transaction Control Language

- Indica o começar de uma transação
 - `BEGIN TRANSACTION <transaction_name> ;`
- Se tudo estiver em ordem com todas as declarações dentro de uma única transação, todas as alterações são registadas em conjunto na base de dados
 - `COMMIT;`
- Se ocorrer algum erro com qualquer uma das declarações agrupadas, todas as alterações são abortadas
 - `ROLLBACK;`

DQL - Data Query Language

- Selecionar dados de uma tabela
 - `SELECT <nome atributo>, <nome atributo 2>, ... FROM <nome tabela>
WHERE <condicao>;`

DCL – Data Control Language

- Para conceder privilégios a uma conta de utilizador, é utilizada declaração GRANT
 - GRANT <privileges_names> ON <object> TO <user>;
- Para remover permissões aos utilizadores é utilizada a declaração REVOKE
 - revoke <privilege_name> on <object_name> from {user_name | public | role_name}

Microsoft SQL Server

Microsoft SQL Server

- Microsoft SQL Server
 - Database Engine
 - Analysis Services
 - Reporting Services
 - SQL Server Graph Databases
 - SQL Server Machine Learning Services
- Microsoft SQL Server – Windows
- Microsoft SQL Server 2017 – Linux/macOS



Microsoft®
SQL Server®

SQL Server vs MySQL

SQL Server vs MySQL

Semelhanças

SQL Server vs MySQL - Semelhanças

- Tabelas
 - Tanto uma como outra utilizam o modelo padrão de tabela de base de dados relacional para armazenar dados em colunas e linhas.
- Alto Desempenho
 - A sua base de dados é a espinha dorsal das suas aplicações, responsável por armazenar e devolver os dados o mais rapidamente possível. Tanto o MySQL como o MSSQL oferecem alta velocidade de desempenho.

SQL Server vs MySQL - Semelhanças

- Chaves
 - Ambas as plataformas usam chaves primárias e estrangeiras para estabelecer as relações entre tabelas.
- Popularidade Web
 - Com exceção da Oracle, MySQL e MSSQL são as bases de dados mais comuns para utilização com aplicações web. Quando se inscreve para hospedagem, é-lhe dada a escolha entre MSSQL e MySQL.

SQL Server vs MySQL - Semelhanças

- Escalabilidade
 - Ambas as plataformas podem facilmente escalar. São adequados para pequenos e grandes projetos e podem suportar milhões de transações todos os dias.
- Sintaxe
 - A sintaxe destas duas plataformas de base de dados é semelhante, apenas com algumas pequenas diferenças entre as diferentes declarações CRUD (criar, ler, atualizar, excluir).

SQL Server vs MySQL - Semelhanças

- Drivers
 - Existem *drivers* para quase todas as linguagem de programação. Isto permite conectar facilmente tanto ao MySQL como ao MSSQL sem a necessidade de codificação específica.

SQL Server vs MySQL

Diferenças

SQL Server vs MySQL – Diferenças

- Compatibilidade

- O SQL Server foi originalmente desenvolvido exclusivamente para o sistema operativo Windows pela Microsoft.
- A Microsoft disponibilizou recentemente RDBMS tanto no Mac OS X como no Linux. Isto dá às empresas a opção de executar o sistema de base de dados em três plataformas distintas.
- Infelizmente, os utilizadores ainda não têm a opção de utilizar certas capacidades enquanto executam o SQL Server no Mac OS X ou Linux.
- O MySQL, por outro lado, pode funcionar sem problemas em vários sistemas operativos, incluindo Linux, Mac OS X e Windows.

SQL Server vs MySQL – Diferenças

- Suporte
 - Tanto o MySQL como o MSSQL suportam múltiplas linguagens de programação, incluindo Java, C++, PHP, Ruby, Python, Delphi, Visual Basic, Go e R.
 - Como o MySQL é tão versátil no seu suporte a linguagens de programação, é popular entre muitas comunidades entre elas a de *open-source*.
 - Embora se utilize os dois tipos de base de dados para projetos Windows e Linux, o MySQL trabalha nativamente com PHP, e o MSSQL é usado principalmente com .NET.
 - A integração é mais simples utilizando o MySQL para PHP e MSSQL para projetos windows.

SQL Server vs MySQL – Diferenças

- MyISAM e InnoDB
 - Estes motores são configurações para o MySQL, que permitem ao programador realizar uma série de atividades de design e programação.
 - O MSSQL não permite especificar diferentes motores quando cria uma base de dados.

SQL Server vs MySQL – Diferenças

- Custo-Benefício

- O MySQL é gratuito e de código aberto, embora tenha de pagar pelo apoio se o exigir.
- O MSSQL é mais caro de executar, porque vai precisar de licenças para o servidor que executa o software.

SQL Server vs MySQL – Diferenças

- LINQ

- O MSSQL permite-lhe configurar as suas classes-quadro de entidade em .NET, o que significa que pode começar com consultas LINQ.
- Com o MySQL, você precisaria de baixar ferramentas de terceiros se você quiser usar .NET.

SQL Server vs MySQL – Diferenças

- **IDE**

- Tanto o MySQL como o MSSQL possuem ferramentas IDE.
- MySQL tem o MySQL Workbench e o MSSQL usa Management Studio.
- Estas ferramentas permitem uma conexão com o servidor de forma a gerir configurações de arquitetura, segurança e desenho de tabelas.

SQL Server vs MySQL – Diferenças

- Coleções Binárias

- MSSQL e MySQL são projetados como coleções binárias.
- MySQL permite que os programadores utilizem binários para manipular ficheiros de base de dados mesmo durante a execução.
- Os ficheiros de base de dados também podem ser acedidos e manipulados por processos alternativos em execução.
- Inversamente, o MSSQL não permite que qualquer processo manipule ou aceda a binários ou ficheiros de base de dados.
- Isto elimina a oportunidade para os hackers acederem ou manipularem diretamente os dados.
- Como tal, o MSSQL é mais seguro do que o MySQL.

SQL Server vs MySQL – Diferenças

- Backup
 - Com o MySQL, o *backup* de dados é feito como declarações SQL.
 - Esta funcionalidade minimiza a possibilidade de corrupção de dados ocorrer enquanto alterna entre versões MySQL ou edições.
 - A desvantagem é que isto torna a restauração de dados um processo demorado, porque requer a execução de várias declarações SQL.
 - O MSSQL não bloqueia a base de dados enquanto faz o backup dos dados, permitindo aos utilizadores fazer *backup* e restaurar quantidades massivas de dados com o mínimo de esforço.

SQL Server vs MySQL – Diferenças

- Stop Query Execution
 - O MySQL não permite que os utilizadores matem ou cancelem uma consulta uma vez que esta começa a funcionar.
 - Para impedir a execução da consulta SQL, os utilizadores têm de matar todo o processo.
 - Os utilizadores do MSSQL podem truncar uma consulta de base de dados enquanto está em execução sem matar todo o processo.
 - Além disso, o MSSQL utiliza um motor transacional para manter um estado consistente.
 - Isto dá ao MSSQL uma vantagem notável sobre o MySQL.

SQL Server vs MySQL

O MSSQL Server foi introduzido em 1989, enquanto o MySQL foi introduzido em 1995 como um projeto de código aberto.

O MySQL pode funcionar em Linux ou Windows. O MSSQL funciona no Windows e geralmente faz parte de um ambiente Windows.

Tanto o MySQL como o MSSQL podem lidar com pequenos e grandes projetos de software, por isso podemos esperar níveis de desempenho semelhantes.

Bibliografia

- B. Gomez,(2020) “Resolviendo problemas de Big Data”, Alfaomega.
- D. Insua, (2019)“Big data: Conceptos, tecnologías y aplicaciones”, CSIC.
- H. Jones, (2019)“Analítica de datos”, HJ,.
- J. Somed, (2020)“Big Data Analytics”, JLC.
- D. Petković (2020)“Microsoft® SQL Server® 2019 A Beginner’s Guide - Seventh Edition”, McGraw Hill.