



# MASTERING RELATIONAL AND NON-RELATIONAL

## 1. Modelagem de Banco de Dados Relacional (10 Pontos)

- Desenvolver o modelo lógico e físico do banco de dados, garantindo que:
- O modelo esteja na **3ª Forma Normal (3FN)**, com todas as cardinalidades corretas.
- Cada entidade e relacionamento estejam coerentes com o tema “**O Futuro do Trabalho**” e com a solução proposta.
- O modelo seja implementado no **banco de dados relacional Oracle**.
- Criar todos os objetos necessários (**tabelas, chaves primárias e estrangeiras, constraints, etc.**).



# MASTERING RELATIONAL AND NON-RELATIONAL

## 2. Procedures e Funções (20 Pontos)

- Criar **procedures** para realizar os *inserts* no banco de dados, utilizando parâmetros e garantindo consistência
- Criar **duas funções personalizadas**, descritas a seguir, aplicadas ao contexto do tema “O Futuro do Trabalho”
  - **Função 1:** responsável por converter dados relacionais em **JSON manualmente construído**, sem uso de funções internas do Oracle.
  - **Função 2:** responsável por **validações e cálculos lógicos**, utilizando expressões regulares e tratamento de exceções.
- Criar **triggers de auditoria** para registrar transações em cada tabela (**INSERT, UPDATE, DELETE**).
- Inserir **no mínimo 10 registros** em cada tabela utilizando as *procedures* de inserção — assegurando que os dados estejam no contexto da aplicação (dados genéricos ou fora do tema serão desconsiderados).
- Implementar **tratamento de exceções e expressões regulares**, conforme boas práticas de **PL/SQL**.
- Além disso:
  - Criar uma **procedure** para exportar um **dataset no formato JSON**, contendo dados relevantes da aplicação (usuários, vagas, competência)
  - Esse dataset poderá ser utilizado para **alimentar uma aplicação de Inteligência Artificial (IA)**, caso o grupo deseje implementar essa inteligência
- **Nota Importante:** Procedures com *hard inserts* ou valores que não pertençam ao contexto do tema “O Futuro do Trabalho” serão desconsideradas.



# MASTERING RELATIONAL AND NON-RELATIONAL

## Sobre a Função 1 — Conversão Manual para JSON (15 Pontos)

- Crie uma **Função** que receba dados relacionais e retorne uma **string no formato JSON**.
- A **lógica de conversão** deve ser desenvolvida integralmente pelo grupo, utilizando concatenação e manipulação para construir o JSON.
- **Proibido o uso de funções internas/built-in do Oracle**, como TO\_JSON, JSON\_OBJECT, JSON\_VALUE, JSON\_ARRAY ou similares.
- A função deve **tratar no mínimo 3 exceções distintas**, com mensagens personalizadas e registradas via DBMS\_OUTPUT em uma tabela de log.
- O JSON gerado deve conter dados coerentes com o contexto da aplicação — por exemplo, **informações de perfil, competências e vagas** relacionadas ao tema *O Futuro do Trabalho*.
- *Exemplo de aplicação*: Gerar um JSON com o perfil de um profissional, suas competências e cursos recomendados para exportado e utilizado posteriormente em uma aplicação de IA ou módulo de recomendação de carreira.



# MASTERING RELATIONAL AND NON-RELATIONAL

- Sobre a Função 2 — Validação e Cálculo de Dados (15 Pontos)
- Crie uma **Função** responsável por realizar **validações ou cálculos lógicos** dentro do contexto da solução.
- Utilize **expressões regulares (REGEXP)** para verificar formatação de campos (exemplo: e-mail corporativo, C etc.).
- Implemente **tratamento de exceções** com mensagens amigáveis, garantindo que o sistema não quebre em entrada ou ausência de dados.
- A lógica deve refletir a proposta do tema “**O Futuro do Trabalho**”, como:
  - Cálculo de **nível de compatibilidade** entre perfil do candidato e competências exigidas em uma vaga;
  - Validação de **dados cadastrais** antes da inserção no banco;
  - Verificação da **aderência de um curso de requalificação** às competências em alta no mercado.
- *Exemplo de aplicação:* Calcular automaticamente o percentual de compatibilidade entre uma vaga, retornando uma mensagem JSON com o resultado da análise.



# MASTERING RELATIONAL AND NON-RELATIONAL

- Empacotamento de Objetos de Banco de Dados (10 Pontos)
- Todos os objetos (**procedures** e **funções**) devem estar **empacotados** para garantir modularidade e reuso do código.
- O empacotamento deve seguir **boas práticas**, agrupando logicamente os objetos conforme suas funções (ex: funções de validação, funções de cálculo, funções de relatório, etc.).



# MASTERING RELATIONAL AND NON-RELATIONAL

- Integração com Outras Linguagens do Curso (Java, C#, Mobile
- A base de dados relacional deverá ser usada como **backend da aplicação**, desenvolvida em Java, C# ou plataforma
- As *procedures* criadas devem ser chamadas pela aplicação e sua execução demonstrada no vídeo de apresentação



# MASTERING RELATIONAL AND NON-RELATIONAL

## • 5. Importação para MongoDB (10 Pontos)

- O dataset exportado em JSON deverá ser **importado para o MongoDB**, criando uma estrutura de dados coerente com o **NoSQL**.
- A estrutura MongoDB deve respeitar os princípios de **organização flexível e eficiente de documentos**, facilitando a integração com APIs ou IA.



# MASTERING RELATIONAL AND NON-RELATIONAL

- Demonstração em Vídeo (10 Pontos)
- Gravar um vídeo demonstrativo apresentando:
  - A execução das *procedures* sendo utilizadas no backend da aplicação.
  - A inserção de dados no banco relacional via *procedures*.
  - A exportação do dataset para JSON e sua importação no MongoDB.
  - Uma explicação de como essa integração contribui para o tema “O Futuro do Trabalho” — mostrando apoiar empresas e pessoas na era da automação, IA e requalificação profissional.





# MASTERING RELATIONAL AND NON-RELATIONAL

## • Entregáveis

- Modelos **Lógico** e **Físico** do banco de dados relacional desenvolvido usando o Oracle Data Modeler (em .pd
- Um arquivo .jpg com o modelo **Lógico** em notação Information Engineering (IE).
- Um arquivo .jpg com o modelo **Físico** em notação Information Engineering (IE).
- **Procedures e Funções** personalizadas com tratamento de exceções (em .sql).
- **Arquivo JSON** gerado a partir do banco relacional.
- **Estrutura MongoDB** (todos os código-fonte usados).
- **Vídeo demonstrativo** com a execução das etapas acima.