

1 - Trabalho Final: Projeto Completo de Banco de Dados MySQL com Relatórios no Jasper Studio

Descrição do Projeto

Você será responsável por criar um sistema completo de gerenciamento de uma biblioteca. O projeto deverá incluir a criação do banco de dados, tabelas, consultas, inserções, atualizações, deleções, e a geração de relatórios utilizando Jasper Studio. Abaixo estão os requisitos detalhados que devem ser implementados para cobrir todos os tópicos abordados durante o curso. Você deverá expandir o modelo proposto criando mais 15 tabelas novas e estabelecendo relacionamentos com as tabelas já existentes no documento. Além da documentação detalhada, você deverá criar o modelo de entidade-relacionamento utilizando o MySQL Workbench.

Estrutura do Banco de Dados

1. Criação do Banco de Dados

- Crie um banco de dados chamado **BibliotecaDB**.

2. Criação das Tabelas

- Crie as seguintes tabelas:
 - **Usuarios:**
 - **usuario_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **data_nascimento** DATE
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
 - **Autores:**
 - **autor_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **nacionalidade** VARCHAR(50)
 - **Editoras:**
 - **editora_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL UNIQUE
 - **cidade** VARCHAR(50)
 - **Categorias:**
 - **categoria_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(50) NOT NULL UNIQUE
 - **Livros:**
 - **livro_id** INT AUTO_INCREMENT PRIMARY KEY
 - **titulo** VARCHAR(200) NOT NULL
 - **isbn** VARCHAR(20) NOT NULL UNIQUE
 - **preco** DECIMAL(10, 2) NOT NULL
 - **data_publicacao** DATE

- **autor_id** INT
- **editora_id** INT
- **categoria_id** INT
- FOREIGN KEY (**autor_id**) REFERENCES **Autores**(**autor_id**)
- FOREIGN KEY (**editora_id**) REFERENCES **Editoras**(**editora_id**)
- FOREIGN KEY (**categoria_id**) REFERENCES **Categorias**(**categoria_id**)
- **Emprestimos:**
 - **emprestimo_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **livro_id** INT
 - **data_emprestimo** DATE NOT NULL
 - **data_devolucao** DATE
 - **data_devolvido** DATE
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios**(**usuario_id**)
 - FOREIGN KEY (**livro_id**) REFERENCES **Livros**(**livro_id**)
- **Reservas:**
 - **reserva_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **livro_id** INT
 - **data_reserva** DATE NOT NULL
 - **data_limite_reserva** DATE
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios**(**usuario_id**)
 - FOREIGN KEY (**livro_id**) REFERENCES **Livros**(**livro_id**)
- **Multas:**
 - **multa_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **emprestimo_id** INT
 - **valor** DECIMAL(10, 2) NOT NULL
 - **pago** BOOLEAN DEFAULT FALSE
 - **data_pagamento** DATE
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios**(**usuario_id**)
 - FOREIGN KEY (**emprestimo_id**) REFERENCES **Emprestimos**(**emprestimo_id**)
- **Avaliacoess:**
 - **avaliacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **livro_id** INT

- **nota** INT CHECK (nota BETWEEN 1 AND 5)
- **comentario** TEXT
- FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- FOREIGN KEY (**livro_id**) REFERENCES **Livros(livro_id)**
- **Funcionarios:**
 - **funcionario_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **data_nascimento** DATE
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
 - **cargo** VARCHAR(50)
 - **salario** DECIMAL(10, 2) NOT NULL
- **Eventos:**
 - **evento_id** INT AUTO_INCREMENT PRIMARY KEY
 - **titulo** VARCHAR(200) NOT NULL
 - **descricao** TEXT
 - **data_inicio** DATE NOT NULL
 - **data_fim** DATE
- **Participacoes:**
 - **participacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **evento_id** INT
 - **data_participacao** DATE
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
 - FOREIGN KEY (**evento_id**) REFERENCES **Eventos(evento_id)**
- **Doacoes:**
 - **doacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **livro_id** INT
 - **data_doacao** DATE NOT NULL
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
 - FOREIGN KEY (**livro_id**) REFERENCES **Livros(livro_id)**
- **Editoras_Contatos:**
 - **editora_id** INT
 - **nome_contato** VARCHAR(100)

- **telefone** VARCHAR(20)
- **email** VARCHAR(100)
- PRIMARY KEY (**editora_id**, **email**)
- FOREIGN KEY (**editora_id**) REFERENCES **Editoras(editora_id)**

3. Inserção de Dados

- Insira pelo menos 20 registros em cada tabela criada.

4. Consultas Básicas

- Crie consultas utilizando **SELECT, WHERE, AND, OR, NOT, ORDER BY, LIMIT**:
 - Exemplo: Liste todos os livros de uma determinada categoria com preço maior que um valor específico.

5. Manipulação de Dados

- Crie comandos para **INSERT INTO, UPDATE, DELETE**:
 - Exemplo: Atualize o preço de um livro específico.
 - Exemplo: Delete todas as reservas que ultrapassaram a data limite.

6. Consultas Avançadas

- Utilize **MIN, MAX, COUNT, AVG, SUM** em suas consultas:
 - Exemplo: Encontre o autor com a maior média de avaliações nos seus livros.

7. Utilização de LIKE, Wildcards, IN, BETWEEN

- Exemplo: Encontre livros cujos títulos começam com uma determinada letra.

8. Aliases e Joins

- Crie consultas utilizando **INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN, Self Join**:
 - Exemplo: Liste todos os empréstimos juntamente com o nome do usuário e o título do livro.

9. UNION e Agrupamento

- Crie consultas utilizando **UNION, GROUP BY, HAVING**:
 - Exemplo: Liste a quantidade de livros emprestados por categoria.

10. Subconsultas e Operadores

- Utilize **EXISTS, ANY, ALL**:
 - Exemplo: Liste os livros que nunca foram reservados.

11. Inserção e Seleção Avançada

- Crie comandos utilizando **INSERT SELECT, CASE**, funções nulas (**IFNULL, COALESCE**), e comentários:
 - Exemplo: Insira registros em uma tabela de resumo de muitas pagas.

12. Gerenciamento do Banco de Dados

- Crie comandos para **Create DB, Drop DB, Create Table, Drop Table, Alter Table**.

13. Constrains

- Utilize **Not Null, Unique, Primary Key, Foreign Key, Check, Default**.

14. Índices e Incrementos

- Crie índices (**Create Index**) e utilize auto incremento (**Auto Increment**).

15. Datas e Visões

- Trabalhe com funções de data e crie visões (**Views**):
 - Exemplo: Crie uma visão para listar todos os empréstimos do mês atual.
-

Relatórios no Jasper Studio

1. Instalação e Configuração

- Instale o Jasper Studio e configure a conexão com o banco de dados **BibliotecaDB**.

2. Criação de Relatórios

- Crie relatórios utilizando Jasper Studio para:
 - Listar todos os livros e suas respectivas categorias.
 - Gerar um relatório de empréstimos mensais.
 - Listar usuários e o número de livros que cada um emprestou.
 - Gerar um relatório de multas pagas por período.
 - Listar eventos e os participantes.

3. Personalização dos Relatórios

- Adicione gráficos e formatações aos relatórios para melhor visualização dos dados.
-

Entrega

O trabalho deverá ser entregue em um arquivo zip contendo:

1. Scripts SQL para criação e manipulação do banco de dados.
2. Arquivo do projeto Jasper Studio com os relatórios criados.
3. Um documento explicando cada etapa do

2 - Trabalho Final: Projeto Completo de Banco de Dados MySQL com Relatórios no Jasper Studio

Descrição do Projeto

Você será responsável por criar um sistema completo de gerenciamento de uma universidade. O projeto deverá incluir a criação do banco de dados, tabelas, consultas, inserções, atualizações, deleções, e a geração de relatórios utilizando Jasper Studio. Abaixo estão os requisitos detalhados que devem ser implementados para cobrir todos os tópicos abordados durante o curso. Você deverá expandir o modelo proposto criando mais 15 tabelas novas e estabelecendo relacionamentos com as tabelas já existentes no documento. Além da documentação detalhada, você deverá criar o modelo de entidade-relacionamento utilizando o MySQL Workbench.

Estrutura do Banco de Dados

1. Criação do Banco de Dados

- Crie um banco de dados chamado **UniversidadeDB**.

2. Criação das Tabelas

- Crie as seguintes tabelas:
 - **Estudantes:**
 - **estudante_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **data_nascimento** DATE
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
 - **Professores:**
 - **professor_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **data_nascimento** DATE
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
 - **departamento** VARCHAR(100)
 - **salario** DECIMAL(10, 2) NOT NULL
 - **Cursos:**
 - **curso_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **descricao** TEXT
 - **creditos** INT NOT NULL
 - **Departamentos:**
 - **departamento_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL UNIQUE

- **localizacao** VARCHAR(100)
- **Disciplinas:**
 - **disciplina_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **descricao** TEXT
 - **creditos** INT NOT NULL
 - **departamento_id** INT
 - FOREIGN KEY (**departamento_id**) REFERENCES **Departamentos**(**departamento_id**)
- **Turmas:**
 - **turma_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **disciplina_id** INT
 - **professor_id** INT
 - FOREIGN KEY (**disciplina_id**) REFERENCES **Disciplinas**(**disciplina_id**)
 - FOREIGN KEY (**professor_id**) REFERENCES **Professores**(**professor_id**)
- **Matriculas:**
 - **matricula_id** INT AUTO_INCREMENT PRIMARY KEY
 - **estudante_id** INT
 - **turma_id** INT
 - **data_matricula** DATE NOT NULL
 - FOREIGN KEY (**estudante_id**) REFERENCES **Estudantes**(**estudante_id**)
 - FOREIGN KEY (**turma_id**) REFERENCES **Turmas**(**turma_id**)
- **Notas:**
 - **nota_id** INT AUTO_INCREMENT PRIMARY KEY
 - **estudante_id** INT
 - **turma_id** INT
 - **nota** DECIMAL(4, 2) CHECK (nota BETWEEN 0 AND 10)
 - FOREIGN KEY (**estudante_id**) REFERENCES **Estudantes**(**estudante_id**)
 - FOREIGN KEY (**turma_id**) REFERENCES **Turmas**(**turma_id**)
- **Eventos:**
 - **evento_id** INT AUTO_INCREMENT PRIMARY KEY
 - **titulo** VARCHAR(200) NOT NULL
 - **descricao** TEXT
 - **data_inicio** DATE NOT NULL
 - **data_fim** DATE
- **Participacoes:**

- **participacao_id** INT AUTO_INCREMENT PRIMARY KEY
- **estudante_id** INT
- **evento_id** INT
- **data_participacao** DATE
- FOREIGN KEY (**estudante_id**) REFERENCES **Estudantes**(**estudante_id**)
- FOREIGN KEY (**evento_id**) REFERENCES **Eventos**(**evento_id**)
- **Bibliotecas:**
 - **biblioteca_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **localizacao** VARCHAR(100)
- **Livros:**
 - **livro_id** INT AUTO_INCREMENT PRIMARY KEY
 - **titulo** VARCHAR(200) NOT NULL
 - **autor** VARCHAR(100)
 - **isbn** VARCHAR(20) NOT NULL UNIQUE
 - **biblioteca_id** INT
 - FOREIGN KEY (**biblioteca_id**) REFERENCES **Bibliotecas**(**biblioteca_id**)
- **Emprestimos:**
 - **emprestimo_id** INT AUTO_INCREMENT PRIMARY KEY
 - **estudante_id** INT
 - **livro_id** INT
 - **data_emprestimo** DATE NOT NULL
 - **data_devolucao** DATE
 - **data_devolvido** DATE
 - FOREIGN KEY (**estudante_id**) REFERENCES **Estudantes**(**estudante_id**)
 - FOREIGN KEY (**livro_id**) REFERENCES **Livros**(**livro_id**)
- **Multas:**
 - **multa_id** INT AUTO_INCREMENT PRIMARY KEY
 - **emprestimo_id** INT
 - **valor** DECIMAL(10, 2) NOT NULL
 - **pago** BOOLEAN DEFAULT FALSE
 - **data_pagamento** DATE
 - FOREIGN KEY (**emprestimo_id**) REFERENCES **Emprestimos**(**emprestimo_id**)
- **Avaliacoes_Cursos:**
 - **avaliacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **estudante_id** INT

- **curso_id** INT
- **nota** INT CHECK (nota BETWEEN 1 AND 5)
- **comentario** TEXT
- FOREIGN KEY (**estudante_id**) REFERENCES **Estudantes**(**estudante_id**)
- FOREIGN KEY (**curso_id**) REFERENCES **Cursos**(**curso_id**)

3. Inserção de Dados

- Insira pelo menos 20 registros em cada tabela criada.

4. Consultas Básicas

- Crie consultas utilizando **SELECT, WHERE, AND, OR, NOT, ORDER BY, LIMIT**:
 - Exemplo: Liste todos os estudantes matriculados em uma determinada turma.

5. Manipulação de Dados

- Crie comandos para **INSERT INTO, UPDATE, DELETE**:
 - Exemplo: Atualize a nota de um estudante em uma turma específica.
 - Exemplo: Delete todas as participações de estudantes em eventos que já passaram.

6. Consultas Avançadas

- Utilize **MIN, MAX, COUNT, AVG, SUM** em suas consultas:
 - Exemplo: Encontre o professor com a maior média de avaliações dos cursos que ele leciona.

7. Utilização de LIKE, Wildcards, IN, BETWEEN

- Exemplo: Encontre estudantes cujos nomes começam com uma determinada letra.

8. Aliases e Joins

- Crie consultas utilizando **INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN, Self Join**:
 - Exemplo: Liste todos os empréstimos juntamente com o nome do estudante e o título do livro.

9. UNION e Agrupamento

- Crie consultas utilizando **UNION, GROUP BY, HAVING**:
 - Exemplo: Liste a quantidade de estudantes matriculados por curso.

10. Subconsultas e Operadores

- Utilize **EXISTS, ANY, ALL**:
 - Exemplo: Liste os cursos que nunca receberam avaliações.

11. Inserção e Seleção Avançada

- Crie comandos utilizando **INSERT SELECT, CASE**, funções nulas (**IFNULL, COALESCE**), e comentários:
 - Exemplo: Insira registros em uma tabela de resumo de multas pagas.

12. Gerenciamento do Banco de Dados

- Crie comandos para **Create DB, Drop DB, Create Table, Drop Table, Alter Table**.

13. Constrains

- Utilize **Not Null, Unique, Primary Key, Foreign Key, Check, Default**.

14. Indices e Incrementos

- Crie índices (**Create Index**) e utilize auto incremento (**Auto Increment**).

15. Datas e Visões

- Trabalhe com funções de data e crie visões (**Views**):
 - Exemplo: Crie uma visão para listar todos os empréstimos do mês atual.
-

Relatórios no Jasper Studio

1. Instalação e Configuração

- Instale o Jasper Studio e configure a conexão com o banco de dados **UniversidadeDB**.

2. Criação de Relatórios

- Crie relatórios utilizando Jasper Studio para:
 - Listar todos os estudantes e os cursos em que estão matriculados.
 - Gerar um relatório de notas por turma.
 - Listar professores e as disciplinas que lecionam.
 - Gerar um relatório de eventos e participações.
 - Listar livros emprestados e suas respectivas multas.

3. Personalização dos Relatórios

- Adicione gráficos

3 - Trabalho Final: Projeto Completo de Banco de Dados MySQL com Relatórios no Jasper Studio

Descrição do Projeto

Você será responsável por criar um sistema completo de gerenciamento de um hospital. O projeto deverá incluir a criação do banco de dados, tabelas, consultas, inserções, atualizações, deleções, e a geração de relatórios utilizando Jasper Studio. Abaixo estão os requisitos detalhados que devem ser implementados para cobrir todos os tópicos abordados durante o curso. Você deverá expandir o modelo proposto criando mais 15 tabelas novas e estabelecendo relacionamentos com as tabelas já existentes no documento. Além da documentação detalhada, você deverá criar o modelo de entidade-relacionamento utilizando o MySQL Workbench.

Estrutura do Banco de Dados

1. Criação do Banco de Dados

- Crie um banco de dados chamado **HospitalDB**.

2. Criação das Tabelas

- Crie as seguintes tabelas:
 - **Pacientes:**
 - **paciente_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **data_nascimento** DATE
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **Medicos:**
 - **medico_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **especialidade** VARCHAR(50)
 - **crm** VARCHAR(20) NOT NULL UNIQUE
 - **telefone** VARCHAR(20)
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **salario** DECIMAL(10, 2) NOT NULL
 - **Enfermeiros:**
 - **enfermeiro_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **corem** VARCHAR(20) NOT NULL UNIQUE
 - **telefone** VARCHAR(20)
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **salario** DECIMAL(10, 2) NOT NULL
 - **Departamentos:**
 - **departamento_id** INT AUTO_INCREMENT PRIMARY KEY

- **nome** VARCHAR(100) NOT NULL UNIQUE
- **localizacao** VARCHAR(100)
- **Salas:**
 - **sala_id** INT AUTO_INCREMENT PRIMARY KEY
 - **numero** VARCHAR(10) NOT NULL
 - **departamento_id** INT
 - **tipo** VARCHAR(50)
 - FOREIGN KEY (**departamento_id**) REFERENCES **Departamentos**(**departamento_id**)
- **Internacoes:**
 - **internacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **paciente_id** INT
 - **sala_id** INT
 - **data_internacao** DATE NOT NULL
 - **data_alta** DATE
 - FOREIGN KEY (**paciente_id**) REFERENCES **Pacientes**(**paciente_id**)
 - FOREIGN KEY (**sala_id**) REFERENCES **Salas**(**sala_id**)
- **Consultas:**
 - **consulta_id** INT AUTO_INCREMENT PRIMARY KEY
 - **paciente_id** INT
 - **medico_id** INT
 - **data_consulta** DATE NOT NULL
 - **descricao** TEXT
 - FOREIGN KEY (**paciente_id**) REFERENCES **Pacientes**(**paciente_id**)
 - FOREIGN KEY (**medico_id**) REFERENCES **Medicos**(**medico_id**)
- **Tratamentos:**
 - **tratamento_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **descricao** TEXT
- **Medicamentos:**
 - **medicamento_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **descricao** TEXT
 - **preco** DECIMAL(10, 2) NOT NULL
- **Prescricoes:**
 - **prescricao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **consulta_id** INT

- **medicamento_id** INT
- **quantidade** INT NOT NULL
- **dosagem** VARCHAR(50)
- FOREIGN KEY (**consulta_id**) REFERENCES **Consultas**(**consulta_id**)
- FOREIGN KEY (**medicamento_id**) REFERENCES **Medicamentos**(**medicamento_id**)
- **Tratamentos_Pacientes:**
 - **tratamento_paciente_id** INT AUTO_INCREMENT PRIMARY KEY
 - **paciente_id** INT
 - **tratamento_id** INT
 - **data_inicio** DATE NOT NULL
 - **data_fim** DATE
 - FOREIGN KEY (**paciente_id**) REFERENCES **Pacientes**(**paciente_id**)
 - FOREIGN KEY (**tratamento_id**) REFERENCES **Tratamentos**(**tratamento_id**)
- **Horarios_Trabalho:**
 - **horario_id** INT AUTO_INCREMENT PRIMARY KEY
 - **medico_id** INT
 - **enfermeiro_id** INT
 - **dia_semana** VARCHAR(10)
 - **hora_inicio** TIME
 - **hora_fim** TIME
 - FOREIGN KEY (**medico_id**) REFERENCES **Medicos**(**medico_id**)
 - FOREIGN KEY (**enfermeiro_id**) REFERENCES **Enfermeiros**(**enfermeiro_id**)
- **Consultas_Historico:**
 - **consulta_historico_id** INT AUTO_INCREMENT PRIMARY KEY
 - **paciente_id** INT
 - **medico_id** INT
 - **data_consulta** DATE NOT NULL
 - **descricao** TEXT
 - **data_criacao_registro** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**paciente_id**) REFERENCES **Pacientes**(**paciente_id**)
 - FOREIGN KEY (**medico_id**) REFERENCES **Medicos**(**medico_id**)

3. Inserção de Dados

- Insira pelo menos 20 registros em cada tabela criada.

4. Consultas Básicas

- Crie consultas utilizando **SELECT, WHERE, AND, OR, NOT, ORDER BY, LIMIT:**
 - Exemplo: Liste todas as consultas de um determinado paciente com um médico específico.

5. Manipulação de Dados

- Crie comandos para **INSERT INTO, UPDATE, DELETE**:
 - Exemplo: Atualize o salário de um enfermeiro.
 - Exemplo: Delete todas as prescrições de medicamentos que não são mais utilizados.

6. Consultas Avançadas

- Utilize **MIN, MAX, COUNT, AVG, SUM** em suas consultas:
 - Exemplo: Encontre o médico com o maior número de consultas realizadas.

7. Utilização de LIKE, Wildcards, IN, BETWEEN

- Exemplo: Encontre pacientes cujos nomes começam com uma determinada letra.

8. Aliases e Joins

- Crie consultas utilizando **INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN, Self Join**:
 - Exemplo: Liste todas as internações juntamente com o nome do paciente e o número da sala.

9. UNION e Agrupamento

- Crie consultas utilizando **UNION, GROUP BY, HAVING**:
 - Exemplo: Liste a quantidade de consultas realizadas por departamento.

10. Subconsultas e Operadores

- Utilize **EXISTS, ANY, ALL**:
 - Exemplo: Liste os pacientes que nunca foram internados.

11. Inserção e Seleção Avançada

- Crie comandos utilizando **INSERT SELECT, CASE**, funções nulas (**IFNULL, COALESCE**), e comentários:
 - Exemplo: Insira registros em uma tabela de resumo de tratamentos realizados.

12. Gerenciamento do Banco de Dados

- Crie comandos para **Create DB, Drop DB, Create Table, Drop Table, Alter Table**.

13. Constrains

- Utilize **Not Null, Unique, Primary Key, Foreign Key, Check, Default**.

14. Índices e Incrementos

- Crie índices (**Create Index**) e utilize auto incremento (**Auto Increment**).

15. Datas e Visões

- Trabalhe com funções de data e crie visões (**Views**):
 - Exemplo: Crie uma visão para listar todas as consultas do mês atual.

Relatórios no Jasper Studio

1. Instalação e Configuração

- Instale o Jasper Studio e configure a conexão com o banco de dados **HospitalDB**.

2. Criação de Relatórios

- Crie relatórios utilizando Jasper Studio para:

- Listar todos os pacientes e suas respectivas internações.
- Gerar um relatório de consultas realizadas por médico.
- Listar médicos e suas especialidades.
- Gerar um relatório de tratamentos e pacientes.
- Listar medicamentos prescritos e suas quantidades.

3. Personalização dos Relatórios

- Adicione gráficos e formatações aos relatórios para melhor visualização dos dados.
-

Entrega

O trabalho deverá ser entregue em um arquivo zip contendo:

1. Scripts SQL para criação e manipulação do banco de dados.
2. Arquivo do projeto Jasper Studio com os relatórios criados.
3. Um documento explicando cada etapa do trabalho e como executar os scripts e visualizar os relatórios.

Avaliação

A avaliação será baseada na:

1. Correção e completude das queries SQL.
2. Qualidade e organização dos relatórios gerados no Jasper Studio.
3. Documentação clara e detalhada explicando o processo e os resultados.

Bom trabalho e sucesso no desenvolvimento

4 - Trabalho Final: Projeto Completo de Banco de Dados MySQL com Relatórios no Jasper Studio

Descrição do Projeto

Você será responsável por criar um sistema completo de gerenciamento de uma empresa de e-commerce. O projeto deverá incluir a criação do banco de dados, tabelas, consultas, inserções, atualizações, deleções, e a geração de relatórios utilizando Jasper Studio. Abaixo estão os requisitos detalhados que devem ser implementados para cobrir todos os tópicos abordados durante o curso. Você deverá expandir o modelo proposto criando mais 15 tabelas novas e estabelecendo relacionamentos com as tabelas já existentes no documento. Além da documentação detalhada, você deverá criar o modelo de entidade-relacionamento utilizando o MySQL Workbench.

Estrutura do Banco de Dados

1. Criação do Banco de Dados

- Crie um banco de dados chamado **EcommerceDB**.

2. Criação das Tabelas

- Crie as seguintes tabelas:
 - **Clientes:**
 - **cliente_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **data_nascimento** DATE
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
 - **Categorias:**
 - **categoria_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL UNIQUE
 - **descricao** TEXT
 - **Produtos:**
 - **produto_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **descricao** TEXT
 - **preco** DECIMAL(10, 2) NOT NULL
 - **estoque** INT NOT NULL
 - **categoria_id** INT
 - FOREIGN KEY (**categoria_id**) REFERENCES **Categorias**(**categoria_id**)
 - **Pedidos:**
 - **pedido_id** INT AUTO_INCREMENT PRIMARY KEY
 - **cliente_id** INT
 - **data_pedido** DATE NOT NULL
 - **status** VARCHAR(20)

- **valor_total** DECIMAL(10, 2) NOT NULL
- FOREIGN KEY (**cliente_id**) REFERENCES **Clientes**(**cliente_id**)
- **Itens_Pedido:**
 - **item_pedido_id** INT AUTO_INCREMENT PRIMARY KEY
 - **pedido_id** INT
 - **produto_id** INT
 - **quantidade** INT NOT NULL
 - **preco_unitario** DECIMAL(10, 2) NOT NULL
 - FOREIGN KEY (**pedido_id**) REFERENCES **Pedidos**(**pedido_id**)
 - FOREIGN KEY (**produto_id**) REFERENCES **Produtos**(**produto_id**)
- **Pagamentos:**
 - **pagamento_id** INT AUTO_INCREMENT PRIMARY KEY
 - **pedido_id** INT
 - **data_pagamento** DATE NOT NULL
 - **valor** DECIMAL(10, 2) NOT NULL
 - **metodo_pagamento** VARCHAR(50)
 - FOREIGN KEY (**pedido_id**) REFERENCES **Pedidos**(**pedido_id**)
- **Fornecedores:**
 - **fornecedor_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **contato** VARCHAR(100)
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
- **Compras:**
 - **compra_id** INT AUTO_INCREMENT PRIMARY KEY
 - **fornecedor_id** INT
 - **data_compra** DATE NOT NULL
 - **valor_total** DECIMAL(10, 2) NOT NULL
 - FOREIGN KEY (**fornecedor_id**) REFERENCES **Fornecedores**(**fornecedor_id**)
- **Itens_Compra:**
 - **item_compra_id** INT AUTO_INCREMENT PRIMARY KEY
 - **compra_id** INT
 - **produto_id** INT
 - **quantidade** INT NOT NULL
 - **preco_unitario** DECIMAL(10, 2) NOT NULL
 - FOREIGN KEY (**compra_id**) REFERENCES **Compras**(**compra_id**)

- FOREIGN KEY (**produto_id**) REFERENCES **Produtos**(**produto_id**)
- **Transportadoras:**
 - **transportadora_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **contato** VARCHAR(100)
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
- **Envios:**
 - **envio_id** INT AUTO_INCREMENT PRIMARY KEY
 - **pedido_id** INT
 - **transportadora_id** INT
 - **data_envio** DATE NOT NULL
 - **data_entrega** DATE
 - FOREIGN KEY (**pedido_id**) REFERENCES **Pedidos**(**pedido_id**)
 - FOREIGN KEY (**transportadora_id**) REFERENCES **Transportadoras**(**transportadora_id**)
- **Avaliacoes:**
 - **avaliacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **cliente_id** INT
 - **produto_id** INT
 - **nota** INT CHECK (nota BETWEEN 1 AND 5)
 - **comentario** TEXT
 - **data_avaliacao** DATE
 - FOREIGN KEY (**cliente_id**) REFERENCES **Clientes**(**cliente_id**)
 - FOREIGN KEY (**produto_id**) REFERENCES **Produtos**(**produto_id**)
- **Cupons:**
 - **cupom_id** INT AUTO_INCREMENT PRIMARY KEY
 - **codigo** VARCHAR(20) NOT NULL UNIQUE
 - **desconto** DECIMAL(5, 2) NOT NULL
 - **data_validade** DATE NOT NULL
- **Pedidos_Cupons:**
 - **pedido_cupom_id** INT AUTO_INCREMENT PRIMARY KEY
 - **pedido_id** INT
 - **cupom_id** INT
 - FOREIGN KEY (**pedido_id**) REFERENCES **Pedidos**(**pedido_id**)
 - FOREIGN KEY (**cupom_id**) REFERENCES **Cupons**(**cupom_id**)
- **Estoque:**

- **estoque_id** INT AUTO_INCREMENT PRIMARY KEY
- **produto_id** INT
- **quantidade** INT NOT NULL
- **localizacao** VARCHAR(100)
- FOREIGN KEY (**produto_id**) REFERENCES **Produtos**(**produto_id**)
- **Devolucoes:**
 - **devolucao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **pedido_id** INT
 - **produto_id** INT
 - **data_devolucao** DATE NOT NULL
 - **motivo** TEXT
 - **quantidade** INT NOT NULL
 - FOREIGN KEY (**pedido_id**) REFERENCES **Pedidos**(**pedido_id**)
 - FOREIGN KEY (**produto_id**) REFERENCES **Produtos**(**produto_id**)
- **Categorias_Pais:**
 - **categoria_pais_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL UNIQUE
 - **descricao** TEXT
- **Categorias_Subcategorias:**
 - **categoria_id** INT
 - **categoria_pais_id** INT
 - PRIMARY KEY (**categoria_id**, **categoria_pais_id**)
 - FOREIGN KEY (**categoria_id**) REFERENCES **Categorias**(**categoria_id**)
 - FOREIGN KEY (**categoria_pais_id**) REFERENCES **Categorias_Pais**(**categoria_pais_id**)

3. Inserção de Dados

- Insira pelo menos 20 registros em cada tabela criada.

4. Consultas Básicas

- Crie consultas utilizando **SELECT, WHERE, AND, OR, NOT, ORDER BY, LIMIT**:
 - Exemplo: Liste todos os produtos de uma determinada categoria com preço maior que um valor específico.

5. Manipulação de Dados

- Crie comandos para **INSERT INTO, UPDATE, DELETE**:
 - Exemplo: Atualize o preço de um produto específico.
 - Exemplo: Delete todos os pedidos que estão com status "Cancelado".

6. Consultas Avançadas

- Utilize **MIN, MAX, COUNT, AVG, SUM** em suas consultas:

- Exemplo: Encontre o produto mais vendido.

7. Utilização de LIKE, Wildcards, IN, BETWEEN

- Exemplo: Encontre clientes cujos nomes começam com uma determinada letra.

8. Aliases e Joins

- Crie consultas utilizando **INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN, Self Join**:
 - Exemplo: Liste todos os pedidos juntamente com o nome do cliente e o status do pedido.

9. UNION e Agrupamento

- Crie consultas utilizando **UNION, GROUP BY, HAVING**:
 - Exemplo: Liste a quantidade de produtos vendidos por categoria.

10. Subconsultas e Operadores

- Utilize **EXISTS, ANY, ALL**:
 - Exemplo: Liste os produtos que nunca receberam avaliações.

11. Inserção e Seleção Avançada

- Crie comandos utilizando **INSERT SELECT, CASE**, funções nulas (**IFNULL, COALESCE**), e comentários:
 - Exemplo: Insira registros em uma tabela de resumo de devoluções.

12. Gerenciamento do Banco de Dados

- Crie comandos para **Create DB, Drop DB, Create Table, Drop Table, Alter Table**.

13. Constrains

- Utilize **Not Null, Unique, Primary Key, Foreign Key, Check, Default**.

14. Índices e Incrementos

- Crie índices (**Create Index**) e utilize auto incremento (**Auto Increment**).

15. Datas e Visões

- Trabalhe com funções de data e crie visões (**Views**):
 - Exemplo: Crie uma visão para listar todas as vendas do mês atual.

Relatórios no Jasper Studio

1. Instalação e Configuração

- Instale o Jasper Studio e configure a conexão com o banco de dados **EcommerceDB**.

2. Criação de Relatórios

- Crie relatórios utilizando Jasper Studio para:
 - Listar todos os produtos e suas respectivas categorias.
 - Gerar um relatório de vendas mensais.
 - Listar clientes e o número de pedidos que cada um realizou.
 - Gerar um relatório de devoluções por período.
 - Listar avaliações de produtos e suas notas médias.

3. Personalização dos Relatórios

- Adicione gráficos e formatações aos relatórios para melhor visualização dos dados.
-

Entrega

O trabalho deverá ser entregue em um arquivo zip contendo:

1. Scripts SQL para criação e manipulação do banco de dados.
2. Arquivo do projeto Jasper Studio com os relatórios criados.
3. Um documento explicando cada etapa do trabalho e como executar os scripts e visualizar os relatórios.

Avaliação

A avaliação será baseada na:

1. Correção e completude das queries SQL.
2. Qualidade e organização dos relatórios gerados no Jasper Studio.
3. Documentação clara e detalhada explicando o processo e os resultados.

Bom trabalho e sucesso no desenvolvimento do projeto!

5 - Trabalho Final: Projeto Completo de Banco de Dados MySQL com Relatórios no Jasper Studio

Descrição do Projeto

Você será responsável por criar um sistema completo de gerenciamento de uma plataforma de ensino online. O projeto deverá incluir a criação do banco de dados, tabelas, consultas, inserções, atualizações, deleções, e a geração de relatórios utilizando Jasper Studio. Abaixo estão os requisitos detalhados que devem ser implementados para cobrir todos os tópicos abordados durante o curso. Você deverá expandir o modelo proposto criando mais 15 tabelas novas e estabelecendo relacionamentos com as tabelas já existentes no documento. Além da documentação detalhada, você deverá criar o modelo de entidade-relacionamento utilizando o MySQL Workbench.

Estrutura do Banco de Dados

1. Criação do Banco de Dados

- Crie um banco de dados chamado **EnsinoOnlineDB**.

2. Criação das Tabelas

- Crie as seguintes tabelas:
 - **Usuarios:**
 - **usuario_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **senha** VARCHAR(100) NOT NULL
 - **data_nascimento** DATE
 - **endereço** VARCHAR(255)
 - **telefone** VARCHAR(20)
 - **tipo** ENUM('Aluno', 'Instrutor', 'Administrador') NOT NULL
 - **Cursos:**
 - **curso_id** INT AUTO_INCREMENT PRIMARY KEY
 - **titulo** VARCHAR(200) NOT NULL
 - **descricao** TEXT
 - **preço** DECIMAL(10, 2) NOT NULL
 - **data_criacao** DATE NOT NULL
 - **instrutor_id** INT
 - FOREIGN KEY (**instrutor_id**) REFERENCES **Usuarios**(**usuario_id**)
 - **Modulos:**
 - **modulo_id** INT AUTO_INCREMENT PRIMARY KEY
 - **curso_id** INT
 - **titulo** VARCHAR(200) NOT NULL
 - **descricao** TEXT
 - **ordem** INT NOT NULL
 - FOREIGN KEY (**curso_id**) REFERENCES **Cursos**(**curso_id**)

- **Aulas:**
 - **aula_id** INT AUTO_INCREMENT PRIMARY KEY
 - **modulo_id** INT
 - **titulo** VARCHAR(200) NOT NULL
 - **descricao** TEXT
 - **duracao** TIME NOT NULL
 - **video_url** VARCHAR(255)
 - **ordem** INT NOT NULL
 - FOREIGN KEY (**modulo_id**) REFERENCES **Modulos(modulo_id)**
- **Matriculas:**
 - **matricula_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **curso_id** INT
 - **data_matricula** DATE NOT NULL
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
 - FOREIGN KEY (**curso_id**) REFERENCES **Cursos(curso_id)**
- **Avaliacoes:**
 - **avaliacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **curso_id** INT
 - **usuario_id** INT
 - **nota** INT CHECK (nota BETWEEN 1 AND 5)
 - **comentario** TEXT
 - **data_avaliacao** DATE
 - FOREIGN KEY (**curso_id**) REFERENCES **Cursos(curso_id)**
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Pagamentos:**
 - **pagamento_id** INT AUTO_INCREMENT PRIMARY KEY
 - **matricula_id** INT
 - **data_pagamento** DATE NOT NULL
 - **valor** DECIMAL(10, 2) NOT NULL
 - **metodo_pagamento** VARCHAR(50)
 - FOREIGN KEY (**matricula_id**) REFERENCES **Matriculas(matricula_id)**
- **Certificados:**
 - **certificado_id** INT AUTO_INCREMENT PRIMARY KEY
 - **matricula_id** INT
 - **data_emissao** DATE NOT NULL

- **codigo_validacao** VARCHAR(100) UNIQUE NOT NULL
- FOREIGN KEY (**matricula_id**) REFERENCES **Matriculas**(**matricula_id**)
- **Comentarios:**
 - **comentario_id** INT AUTO_INCREMENT PRIMARY KEY
 - **aula_id** INT
 - **usuario_id** INT
 - **comentario** TEXT NOT NULL
 - **data_comentario** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**aula_id**) REFERENCES **Aulas**(**aula_id**)
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios**(**usuario_id**)
- **Categorias:**
 - **categoria_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL UNIQUE
 - **descricao** TEXT
- **Cursos_Categorias:**
 - **curso_id** INT
 - **categoria_id** INT
 - PRIMARY KEY (**curso_id**, **categoria_id**)
 - FOREIGN KEY (**curso_id**) REFERENCES **Cursos**(**curso_id**)
 - FOREIGN KEY (**categoria_id**) REFERENCES **Categorias**(**categoria_id**)
- **Cupons:**
 - **cupom_id** INT AUTO_INCREMENT PRIMARY KEY
 - **codigo** VARCHAR(20) NOT NULL UNIQUE
 - **desconto** DECIMAL(5, 2) NOT NULL
 - **data_validade** DATE NOT NULL
- **Cursos_Cupons:**
 - **curso_cupom_id** INT AUTO_INCREMENT PRIMARY KEY
 - **curso_id** INT
 - **cupom_id** INT
 - FOREIGN KEY (**curso_id**) REFERENCES **Cursos**(**curso_id**)
 - FOREIGN KEY (**cupom_id**) REFERENCES **Cupons**(**cupom_id**)
- **Logs_Acesso:**
 - **log_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **data_acesso** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - **endereco_ip** VARCHAR(50)

- FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Anuncios:**
 - **anuncio_id** INT AUTO_INCREMENT PRIMARY KEY
 - **titulo** VARCHAR(200) NOT NULL
 - **descricao** TEXT
 - **data_publicacao** DATE NOT NULL
 - **usuario_id** INT
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Aulas_Assistidas:**
 - **aula_assistida_id** INT AUTO_INCREMENT PRIMARY KEY
 - **aula_id** INT
 - **usuario_id** INT
 - **data_assistencia** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**aula_id**) REFERENCES **Aulas(aula_id)**
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Tickets_Suporte:**
 - **ticket_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **titulo** VARCHAR(200) NOT NULL
 - **descricao** TEXT
 - **status** ENUM('Aberto', 'Em andamento', 'Fechado') NOT NULL
 - **data_abertura** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - **data_fechamento** TIMESTAMP
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Respostas_Tickets:**
 - **resposta_id** INT AUTO_INCREMENT PRIMARY KEY
 - **ticket_id** INT
 - **usuario_id** INT
 - **resposta** TEXT NOT NULL
 - **data_resposta** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**ticket_id**) REFERENCES **Tickets_Suporte(ticket_id)**
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**

3. Inserção de Dados

- Insira pelo menos 20 registros em cada tabela criada.

4. Consultas Básicas

- Crie consultas utilizando **SELECT, WHERE, AND, OR, NOT, ORDER BY, LIMIT:**

- Exemplo: Liste todos os cursos de uma determinada categoria com preço maior que um valor específico.

5. Manipulação de Dados

- Crie comandos para **INSERT INTO, UPDATE, DELETE**:
 - Exemplo: Atualize o preço de um curso específico.
 - Exemplo: Delete todos os tickets de suporte que estão com status "Fechado".

6. Consultas Avançadas

- Utilize **MIN, MAX, COUNT, AVG, SUM** em suas consultas:
 - Exemplo: Encontre o curso com a maior média de avaliações.

7. Utilização de LIKE, Wildcards, IN, BETWEEN

- Exemplo: Encontre usuários cujos nomes começam com uma determinada letra.

8. Aliases e Joins

- Crie consultas utilizando **INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN, Self Join**:
 - Exemplo: Liste todos os pagamentos juntamente com o nome do usuário e o título do curso.

9. UNION e Agrupamento

- Crie consultas utilizando **UNION, GROUP BY, HAVING**:
 - Exemplo: Liste a quantidade de cursos vendidos por categoria.

10. Subconsultas e Operadores

- Utilize **EXISTS, ANY, ALL**:
 - Exemplo: Liste os cursos que nunca receberam avaliações.

11. Inserção e Seleção Avançada

- Crie comandos utilizando **INSERT SELECT, CASE**, funções nulas (**IFNULL, COALESCE**), e comentários:
 - Exemplo: Insira registros em uma tabela de resumo de cursos assistidos.

12. Gerenciamento do Banco de Dados

- Crie comandos para **Create DB, Drop DB, Create Table, Drop Table, Alter Table**.

13. Constrains

- Utilize **Not Null, Unique, Primary Key, Foreign Key, Check, Default**.

14. Índices e Incrementos

- Crie índices (**Create Index**) e utilize auto incremento (**Auto Increment**).

15. Datas e Visões

- Trabalhe com funções de data e crie visões (**Views**):
 - Exemplo: Crie uma visão para listar todas as matrículas do mês atual.

Relatórios no Jasper Studio

1. Instalação e Configuração

- Instale o Jasper Studio e configure a conexão com o banco de dados **EnsinoOnlineDB**.

2. Criação de Relatórios

- Crie relatórios utilizando Jasper Studio para:
 - Listar todos os cursos e suas respectivas categorias.
 - Gerar um relatório de matrículas mensais.
 - Listar usuários e o número de cursos em que estão matriculados.
 - Gerar um relatório de avaliações de cursos.
 - Listar tickets de suporte e seus status.

3. Personalização dos Relatórios

- Adicione gráficos e formatações aos relatórios para melhor visualização dos dados.
-

Entrega

O trabalho deverá ser entregue em um arquivo zip contendo:

1. Scripts SQL para criação e manipulação do banco de dados.
2. Arquivo do projeto Jasper Studio com os relatórios criados.
3. Um documento explicando cada etapa do trabalho e como executar os scripts e visualizar os relatórios.

Avaliação

A avaliação será baseada na:

1. Correção e completude das queries SQL.
2. Qualidade e organização dos relatórios gerados no Jasper Studio.
3. Documentação clara e detalhada explicando o processo e os resultados.

Bom trabalho e sucesso no desenvolvimento do projeto!

6 - Trabalho Final: Projeto Completo de Banco de Dados MySQL com Relatórios no Jasper Studio

Descrição do Projeto

Você será responsável por criar um sistema completo de gerenciamento de uma plataforma de rede social. O projeto deverá incluir a criação do banco de dados, tabelas, consultas, inserções, atualizações, deleções, e a geração de relatórios utilizando Jasper Studio. Abaixo estão os requisitos detalhados que devem ser implementados para cobrir todos os tópicos abordados durante o curso. Você deverá expandir o modelo proposto criando mais 15 tabelas novas e estabelecendo relacionamentos com as tabelas já existentes no documento. Além da documentação detalhada, você deverá criar o modelo de entidade-relacionamento utilizando o MySQL Workbench.

Estrutura do Banco de Dados

1. Criação do Banco de Dados

- Crie um banco de dados chamado **RedeSocialDB**.

2. Criação das Tabelas

- Crie as seguintes tabelas:
 - **Usuarios:**
 - **usuario_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **senha** VARCHAR(100) NOT NULL
 - **data_nascimento** DATE
 - **endereco** VARCHAR(255)
 - **telefone** VARCHAR(20)
 - **bio** TEXT
 - **Amizades:**
 - **amizade_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id1** INT
 - **usuario_id2** INT
 - **data_amizade** DATE NOT NULL
 - **status** ENUM('Pendente', 'Aceita', 'Recusada') NOT NULL
 - FOREIGN KEY (**usuario_id1**) REFERENCES **Usuarios(usuario_id)**
 - FOREIGN KEY (**usuario_id2**) REFERENCES **Usuarios(usuario_id)**
 - **Postagens:**
 - **postagem_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **conteudo** TEXT NOT NULL
 - **data_postagem** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
 - **Comentarios:**

- **comentario_id** INT AUTO_INCREMENT PRIMARY KEY
- **postagem_id** INT
- **usuario_id** INT
- **conteudo** TEXT NOT NULL
- **data_comentario** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
- FOREIGN KEY (**postagem_id**) REFERENCES **Postagens(postagem_id)**
- FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Curtidas:**
 - **curtida_id** INT AUTO_INCREMENT PRIMARY KEY
 - **postagem_id** INT
 - **usuario_id** INT
 - **data_curtida** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**postagem_id**) REFERENCES **Postagens(postagem_id)**
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Grupos:**
 - **grupo_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **descricao** TEXT
 - **data_criacao** DATE NOT NULL
- **Membros_Grupos:**
 - **membro_grupo_id** INT AUTO_INCREMENT PRIMARY KEY
 - **grupo_id** INT
 - **usuario_id** INT
 - **data_entrada** DATE NOT NULL
 - **papel** ENUM('Admin', 'Membro') NOT NULL
 - FOREIGN KEY (**grupo_id**) REFERENCES **Grupos(grupo_id)**
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Mensagens:**
 - **mensagem_id** INT AUTO_INCREMENT PRIMARY KEY
 - **remetente_id** INT
 - **destinatario_id** INT
 - **conteudo** TEXT NOT NULL
 - **data_mensagem** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**remetente_id**) REFERENCES **Usuarios(usuario_id)**
 - FOREIGN KEY (**destinatario_id**) REFERENCES **Usuarios(usuario_id)**
- **Eventos:**

- **evento_id** INT AUTO_INCREMENT PRIMARY KEY
- **nome** VARCHAR(200) NOT NULL
- **descricao** TEXT
- **data_evento** DATE NOT NULL
- **local** VARCHAR(200)
- **criador_id** INT
- FOREIGN KEY (**criador_id**) REFERENCES **Usuarios(usuario_id)**
- **Participacoes_Eventos:**
 - **participacao_evento_id** INT AUTO_INCREMENT PRIMARY KEY
 - **evento_id** INT
 - **usuario_id** INT
 - **data_participacao** DATE NOT NULL
 - FOREIGN KEY (**evento_id**) REFERENCES **Eventos(evento_id)**
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Publicidades:**
 - **publicidade_id** INT AUTO_INCREMENT PRIMARY KEY
 - **titulo** VARCHAR(200) NOT NULL
 - **descricao** TEXT
 - **data_inicio** DATE NOT NULL
 - **data_fim** DATE NOT NULL
 - **empresa** VARCHAR(100)
- **Logs_Acesso:**
 - **log_acesso_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **data_acesso** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - **endereco_ip** VARCHAR(50)
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Notificacoes:**
 - **notificacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **conteudo** TEXT NOT NULL
 - **data_notificacao** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - **lida** BOOLEAN DEFAULT FALSE
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Config_Privacidade:**
 - **config_privacidade_id** INT AUTO_INCREMENT PRIMARY KEY

- **usuario_id** INT
- **perfil_visibilidade** ENUM('Publico', 'Amigos', 'Privado') NOT NULL
- **mensagens_visibilidade** ENUM('Todos', 'Amigos', 'Ninguem') NOT NULL
- FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
- **Recomendacoes_Amigos:**
 - **recomendacao_id** INT AUTO_INCREMENT PRIMARY KEY
 - **usuario_id** INT
 - **amigo_recomendado_id** INT
 - **data_recomendacao** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**
 - FOREIGN KEY (**amigo_recomendado_id**) REFERENCES **Usuarios(usuario_id)**
- **Tags_Postagens:**
 - **tag_id** INT AUTO_INCREMENT PRIMARY KEY
 - **nome** VARCHAR(100) NOT NULL
 - **descricao** TEXT
- **Postagens_Tags:**
 - **postagem_id** INT
 - **tag_id** INT
 - PRIMARY KEY (**postagem_id, tag_id**)
 - FOREIGN KEY (**postagem_id**) REFERENCES **Postagens(postagem_id)**
 - FOREIGN KEY (**tag_id**) REFERENCES **Tags_Postagens(tag_id)**
- **Denuncias:**
 - **denuncia_id** INT AUTO_INCREMENT PRIMARY KEY
 - **postagem_id** INT
 - **usuario_id** INT
 - **motivo** TEXT
 - **data_denuncia** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
 - FOREIGN KEY (**postagem_id**) REFERENCES **Postagens(postagem_id)**
 - FOREIGN KEY (**usuario_id**) REFERENCES **Usuarios(usuario_id)**

3. Inserção de Dados

- Insira pelo menos 20 registros em cada tabela criada.

4. Consultas Básicas

- Crie consultas utilizando **SELECT, WHERE, AND, OR, NOT, ORDER BY, LIMIT**:
 - Exemplo: Liste todas as postagens de um determinado usuário que contenham uma palavra específica.

5. Manipulação de Dados

- Crie comandos para **INSERT INTO**, **UPDATE**, **DELETE**:
 - Exemplo: Atualize a descrição de um grupo específico.
 - Exemplo: Delete todas as notificações de um usuário específico que já foram lidas.

6. Consultas Avançadas

- Utilize **MIN**, **MAX**, **COUNT**, **AVG**, **SUM** em suas consultas:
 - Exemplo: Encontre o usuário com o maior número de postagens.

7. Utilização de LIKE, Wildcards, IN, BETWEEN

- Exemplo: Encontre amigos cujos nomes começam com uma determinada letra.

8. Aliases e Joins

- Crie consultas utilizando **INNER JOIN**, **LEFT JOIN**, **RIGHT JOIN**, **CROSS JOIN**, **Self Join**:
 - Exemplo: Liste todas as mensagens juntamente com o nome do remetente e do destinatário.

9. UNION e Agrupamento

- Crie consultas utilizando **UNION**, **GROUP BY**, **HAVING**:
 - Exemplo: Liste a quantidade de postagens por tag.

10. Subconsultas e Operadores

- Utilize **EXISTS**, **ANY**, **ALL**:
 - Exemplo: Liste os grupos que não têm membros.

11. Inserção e Seleção Avançada

- Crie comandos utilizando **INSERT SELECT**, **CASE**, funções nulas (**IFNULL**, **COALESCE**), e comentários:
 - Exemplo: Insira registros em uma tabela de resumo de denúncias.

12. Gerenciamento do Banco de Dados

- Crie comandos para **Create DB**, **Drop DB**, **Create Table**, **Drop Table**, **Alter Table**.

13. Constrains

- Utilize **Not Null**, **Unique**, **Primary Key**, **Foreign Key**, **Check**, **Default**.

14. Índices e Incrementos

- Crie índices (**Create Index**) e utilize auto incremento (**Auto Increment**).

15. Datas e Visões

- Trabalhe com funções de data e crie visões (**Views**):
 - Exemplo: Crie uma visão para listar todas as postagens do mês atual.

Relatórios no Jasper Studio

1. Instalação e Configuração

- Instale o Jasper Studio e configure a conexão com o banco de dados **RedeSocialDB**.

2. Criação de Relatórios

- Crie relatórios utilizando Jasper Studio para:
 - Listar todos os usuários e suas postagens.

- Gerar um relatório de novas amizades por mês.
- Listar grupos e seus membros.
- Gerar um relatório de eventos e participações.
- Listar denúncias e status de resolução.

3. Personalização dos Relatórios

- Adicione gráficos e formatações aos relatórios para melhor visualização dos dados.
-

Entrega

O trabalho deverá ser entregue em um arquivo zip contendo:

1. Scripts SQL para criação e manipulação do banco de dados.
2. Arquivo do projeto Jasper Studio com os relatórios criados.
3. Um documento explicando cada etapa do trabalho e como executar os scripts e visualizar os relatórios.

Avaliação

A avaliação será baseada na:

1. Correção e completude das queries SQL.
2. Qualidade e organização dos relatórios gerados no Jasper Studio.
3. Documentação clara e detalhada explicando o processo e os resultados.

Bom trabalho e sucesso no desenvolvimento do projeto!