

Laboratório de Banco de Dados Avançado



Estudo de Caso

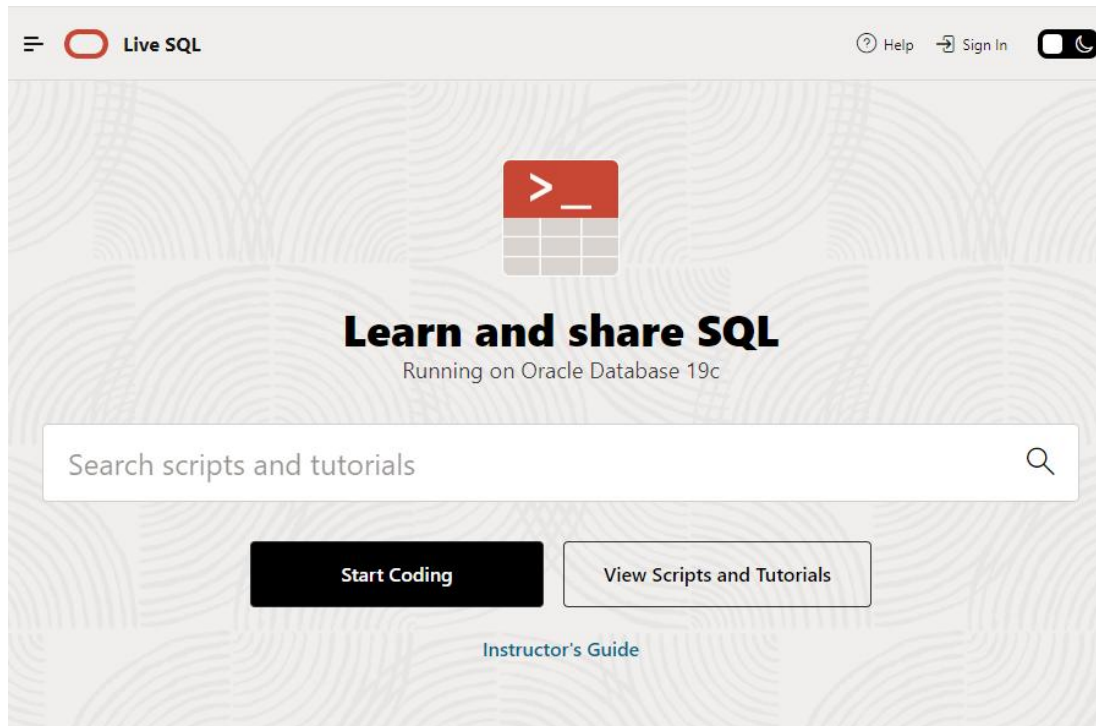
MER – Modelo Entidade Relacionamento



Enunciado

- Criar um banco de dados que faça agendamento de consultas para os pacientes em uma clínica médica.

Acesso ao Live SQL Oracle



- Fazer login no Live SQL da Oracle
- <https://livesql.oracle.com/>

Scripts

```
create table tbEspecialidade(  
    idEspecialidade number primary key,  
    especialidade varchar2(40),  
    deleted number(1) DEFAULT 0,  
    data_criacao date DEFAULT sysdate,  
    data_alteracao date DEFAULT sysdate  
);
```

```
insert into tbEspecialidade (idEspecialidade,especialidade) values(1,'Pediatria');
```

```
select * from tbEspecialidade;
```

```
create table tbMedico(  
    idMedico number primary key,  
    nomeMedico varchar2(40),  
    crmMedico varchar(40),  
    idEspecialidade number,  
    deleted number(1) DEFAULT 0,  
    data_criacao date DEFAULT sysdate,  
    data_alteracao date DEFAULT sysdate,  
    foreign key (idEspecialidade) references tbEspecialidade(idEspecialidade)  
);
```

```
insert into tbMedico (idMedico,nomeMedico,crmMedico,idEspecialidade) values(1,'Moacir','11111',1);
```

```
select * from tbMedico;
```

Scripts

```
create table tbPaciente(  
    idPaciente number primary key,  
    nomePaciente varchar2(40),  
    emailPaciente varchar2(40),  
    cpfPaciente varchar2(14),  
    deleted number(1) DEFAULT 0,  
    data_criacao date DEFAULT sysdate,  
    data_alteracao date DEFAULT sysdate  
);
```

```
insert into tbPaciente (idPaciente,nomePaciente,emailPaciente,cpfPaciente)  
    values(1,'Gabriela','gabriela@gmail.com','111.111.111-11');
```

```
select * from tbPaciente;
```

```
create table tbAgendaConsulta(  
    idAgendaConsulta number primary key,  
    idPaciente number,  
    idMedico number,  
    dataConsulta date,  
    horaConsulta timestamp,  
    deleted number(1) DEFAULT 0,  
    data_criacao date DEFAULT sysdate,  
    data_alteracao date DEFAULT sysdate,  
    foreign key (idPaciente) references tbPaciente(idPaciente),  
    foreign key (idMedico) references tbMedico(idMedico)  
);
```

```
insert into tbAgendaConsulta(  
    idAgendaConsulta,  
    idPaciente,  
    idMedico,  
    dataConsulta,  
    horaConsulta  
)  
values(  
    1,  
    1,  
    1,  
    to_date('2025-08-30','yyyy-mm-dd'),  
    to_timestamp('10:00:00','hh24:mi:ss')  
);
```

```
select * from tbAgendaConsulta;
```



Views

Conceitos de Views



- São tabelas virtuais, resultados de consultas predefinidas;
- Funcionam de forma semelhante a uma tabela. Utilizada em comandos SELECT, INSERT, UPDATE e DELETE;
- Simplifica consultas complexas;
- Controla o acesso a dados sensíveis;
- Permite interações de dados de múltiplas tabelas como se fossem uma única entidade.
- Podem ser consultadas e até modificadas, refletindo sempre as alterações feitas nas tabelas base.

Vantagens e Desvantagens da utilização de Views



Vantagens

- **Simplificação**: Tornam a consulta de dados complexos mais simples.
- **Segurança**: protege informações sensíveis.
- **Flexibilidade**: pode ser alterada sem afetar a estrutura das tabelas base.

Desvantagens

- **Complexidade Oculta**: pode ocultar a complexidade das consultas subjacentes, dificultando a otimização do desempenho por parte dos desenvolvedores.
- **Camada Adicional**: criam mais objetos no banco de dados, aumentando a necessidade de administração.
- **Limitação**: podem limitar o que o usuário pode acessar, impedindo certas tarefas de forma exagerada.

Criação de uma view



```
create view vw_paciente
as
    select nomePaciente as nome, emailPaciente as email from tbPaciente;
```

Execução de uma View



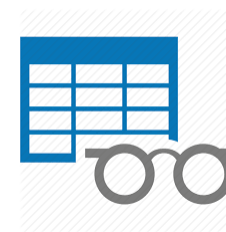
```
select * from vw_paciente;  
select nome from vw_paciente;  
select nome,email from vw_paciente;
```

Exclusão de uma View



```
drop view vw_paciente;
```

Alteração de uma View



```
create or replace view vw_paciente  
as  
    select nomePaciente as nome, emailPaciente as email, cpfPaciente as cpf from tbPaciente;
```

Verificar views



```
desc vw_paciente;  
describe vw_paciente;
```

VIEW VW_PACIENTE

Column	Null?	Type
NOME	-	VARCHAR2(40)
EMAIL	-	VARCHAR2(40)
CPF	-	VARCHAR2(14)

Insert, Update e Delete com a utilização de view



- Funciona somente em views simples (baseadas em uma tabela, sem DISTINCT, GROUP BY, JOIN, UNION, etc.).
- A view precisa incluir as colunas obrigatórias da tabela.

```
create or replace view vw_paciente_simples
as
    select idPaciente, nomePaciente, emailPaciente, cpfPaciente from tbPaciente;

insert into vw_paciente_simples (idPaciente, nomePaciente, emailPaciente, cpfPaciente)
values (2, 'Felipe Farias', 'felipefarias@gmail.com', '222.222.222-22');

update vw_paciente_simples
set emailPaciente = 'gabi@gmail.com'
where idPaciente = 1;

delete from vw_paciente_simples where idPaciente = 2;
```

Exercícios

1. Crie e execute uma view chamada vw_medico_especialidade que mostre o nome do médico, o crm e a especialidade.
2. Crie e execute uma view chamada vw_consulta_paciente que mostre o nome do paciente, o nome do médico, a especialidade do médico, e a data e hora da consulta.
3. Crie e execute uma view com o nome vw_pacientes_ativos que mostre somente os pacientes que não foram "excluídos".
4. Crie e execute uma view com o nome vw_consultas_futuras que traga apenas consultas cuja data da consulta seja maior que o dia atual.
5. Crie e execute uma view com o nome vw_quantidade_pacientes que exiba o total de pacientes cadastrados.
6. Crie e execute uma view com o nome vw_consultas_manha que exiba apenas as consultas agendadas na data atual no período entre às 8h e 12h.

Contato

E-mail: allan@cruzeirosul.edu.br



Universidade **Cruzeiro do Sul**

www.cruzeirodosul.edu.br