

## Cardinalidade

A cardinalidade refere-se à quantidade de valores distintos em uma coluna de uma tabela. Em termos simples, é a medida de "unicidade" dos valores em uma coluna. Entender a cardinalidade é fundamental ao projetar índices, pois afeta diretamente o desempenho das consultas e a eficácia dos índices.

- **Baixa Cardinalidade:** Significa que uma grande parte dos valores em uma coluna é repetida. Por exemplo, uma coluna "status" que só pode ter os valores "ativo" ou "inativo" tem baixa cardinalidade.
- **Alta Cardinalidade:** Indica que uma coluna tem muitos valores únicos. Por exemplo, uma coluna "CPF" em uma tabela de clientes tende a ter alta cardinalidade, já que cada cliente geralmente tem um CPF único.

### Exemplo Prático:

Suponha que temos uma tabela chamada `alunos` com uma coluna `cidade_nascimento` que registra a cidade de nascimento de cada aluno. Vamos considerar dois cenários:

#### Baixa Cardinalidade:

- A maioria dos alunos é de apenas algumas cidades grandes.
- Exemplo: As cidades de São Paulo, Rio de Janeiro e Belo Horizonte são as únicas

idades de nascimento dos alunos.

- Nesse caso, a cardinalidade da coluna cidade\_nascimento é baixa, pois há apenas algumas cidades distintas representadas.

Alta Cardinalidade:

- Cada aluno vem de uma cidade diferente.
- Exemplo: Cada aluno tem uma cidade de nascimento única, e não há cidades repetidas.
- Nesse caso, a cardinalidade da coluna cidade\_nascimento é alta, pois praticamente cada aluno tem uma cidade de nascimento diferente.

-- Criar a tabela de alunos

```
CREATE TABLE alunos (  
  id NUMBER PRIMARY KEY,  
  nome VARCHAR2(100),  
  cidade_nascimento VARCHAR2(100)  
);
```

-- Inserir alguns dados de exemplo para baixa cardinalidade

```
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (1, 'João', 'São Paulo');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (2, 'Maria', 'Rio de Janeiro');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (3, 'Pedro', 'Belo Horizonte');
```

-- Inserir alguns dados de exemplo para alta cardinalidade

```
INSERT INTO alunos (id, nome, cidade_nascimento)
```

```
VALUES (4, 'Ana', 'Curitiba');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (5, 'Carlos', 'Porto Alegre');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (6, 'Mariana', 'Florianópolis');
```

Neste exemplo, a coluna `cidade_nascimento` teria uma baixa cardinalidade para os primeiros três alunos e uma alta cardinalidade para os últimos três alunos. Isso pode influenciar a escolha e a eficácia dos índices ao projetar o esquema de banco de dados e otimizar consultas.

Vamos elaborar uma consulta que busca alunos por cidade de nascimento e, em seguida, criar um índice para melhorar seu desempenho. Além disso, vamos inserir mais dados na tabela para tornar o índice mais eficaz.

```
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (7, 'Lucas', 'São Paulo');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (8, 'Fernanda', 'São Paulo');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (9, 'Marcela', 'Rio de Janeiro');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (10, 'Rafael', 'Belo Horizonte');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (11, 'Aline', 'Belo Horizonte');  
INSERT INTO alunos (id, nome, cidade_nascimento)  
VALUES (12, 'Thiago', 'Curitiba');
```

```
INSERT INTO alunos (id, nome, cidade_nascimento)
VALUES (13, 'Patrícia', 'Curitiba');
INSERT INTO alunos (id, nome, cidade_nascimento)
VALUES (14, 'Camila', 'Porto Alegre');
INSERT INTO alunos (id, nome, cidade_nascimento)
VALUES (15, 'Roberto', 'Porto Alegre');
INSERT INTO alunos (id, nome, cidade_nascimento)
VALUES (16, 'Amanda', 'Florianópolis');
```

Agora, vamos criar uma consulta simples que busca alunos por cidade de nascimento:

```
-- Consulta sem índice
SELECT *
FROM alunos
WHERE cidade_nascimento = 'São Paulo';
```

Para melhorar o desempenho desta consulta, vamos criar um índice na coluna cidade\_nascimento:

```
-- Criar um índice na coluna cidade_nascimento
CREATE INDEX idx_cidade_nascimento ON
alunos(cidade_nascimento);
```

**## Verificando a cardinalidade da coluna**

Para verificar a cardinalidade de uma coluna em uma tabela no Oracle, você pode usar uma consulta que conta os valores distintos na coluna. Aqui está como

você pode fazer isso:

```
SELECT COUNT(DISTINCT coluna) AS cardinalidade  
FROM sua_tabela;
```

Por exemplo, para verificar a cardinalidade da coluna `cidade_nascimento` na tabela `alunos`, você pode executar:

```
SELECT COUNT(DISTINCT cidade_nascimento) AS  
cardinalidade  
FROM alunos;
```

Isso retornará o número de valores distintos na coluna `cidade_nascimento`, que é a cardinalidade dessa coluna.

Como Saber:

Para determinar se a cardinalidade é alta ou baixa, você pode considerar o seguinte:

Número total de linhas na tabela: Primeiro, determine o tamanho da tabela.

```
SELECT COUNT(*) FROM sua_tabela;
```

Número de valores únicos na coluna: Depois, encontre a cardinalidade da coluna como explicado anteriormente.

- Avaliação: Compare o número de valores únicos (cardinalidade) com o número total de linhas na tabela.
  - Se a cardinalidade estiver próxima do número total de linhas, a coluna tem alta cardinalidade.
  - Se a cardinalidade for uma pequena fração do número total de linhas, a coluna tem baixa cardinalidade.

### Exemplo Prático:

Imagine que sua tabela alunos tem 10.000 linhas. Você executa a consulta para contar valores únicos na coluna cidade\_nascimento e encontra 200 valores únicos.

- Número total de linhas: 10.000
- Valores únicos (cardinalidade): 200

Neste caso, a coluna cidade\_nascimento tem uma cardinalidade baixa a média em relação ao número total de linhas, porque os 200 valores únicos são uma pequena fração do total de 10.000 linhas. Se, por outro lado, a coluna contivesse 9.500 valores únicos, consideraríamos isso como uma alta cardinalidade.