

s)

A integridade dos dados no sistema é assegurada não apenas por restrições como PRIMARY KEY, FOREIGN KEY e CHECK, mas também por meio de mecanismos adicionais de controlo lógico, como triggers, procedimentos e validações na aplicação. Estes mecanismos garantem coerência, corretude temporal e respeito pelas regras de negócio.

## **1- Consistência de Atributos Derivados**

PRODUTO.QUANT\_EM\_MAQUINAS – Reflete a soma do STOCK\_COMPARTIMENTO de todos os compartimentos, em todas as máquinas, que armazenam esse produto.

QUANTIDADE\_PRODUTO – Reflete a soma das quantidades armazenadas nos armazéns ou seja, o total da coluna QUANTIDADE\_ARMAZENADA da tabela ARMAZENA, para cada produto.

A tabela PRODUTO também armazena o total de unidades vendidas (UNIDADES\_VENDIDAS), que deve ser incrementado automaticamente sempre que ocorre uma venda.

Estes valores são mantidos atualizados através de triggers do tipo AFTER INSERT, UPDATE ou DELETE, nas tabelas COMPARTIMENTO, VENDA e ARMAZENA, de modo a recalcular os totais e garantir a consistência.

## **2 - Validação de valores e intervalos**

Existem atributos que, por regra de negócio, não podem assumir valores negativos. Por exemplo: PRECO na tabela PRODUTO, STOCK\_COMPARTIMENTO, CAP\_COMPARTIMENTO e QUANTIDADE\_COMPARTIMENTO na tabela COMPARTIMENTO, AUTONOMIA e CAPACIDADE na tabela VEICULO, e DISTANCIA na tabela VIAGEM. Estas validações são feitas com triggers BEFORE INSERT OR UPDATE, que lançam uma exceção se for inserido ou alterado um valor inválido.

Além disso, é necessário validar intervalos de datas. A data de fim de uma viagem (VIAGEM.DATA\_FIM) não pode ser anterior à data de início (VIAGEM.DATA\_INICIO). Datas como DATA\_VISITA, DATA\_VENDA ou DATA\_REPOSICAO não devem estar no futuro. A data de uma encomenda deve ser sempre anterior à data atual (SYSDATE). Estas regras são verificadas também por triggers ou procedimentos, antes das inserções ou atualizações.

### 3 – Regras de Negócio

**Venda de produto:** só pode ocorrer se o compartimento da máquina tiver stock suficiente para o produto em causa. Esta verificação é feita por uma trigger BEFORE INSERT na tabela VENDA, que verifica o stock no COMPARTIMENTO. Após a venda, outra trigger AFTER INSERT atualiza o stock no compartimento, incrementa o número de unidades vendidas no PRODUTO e, se a máquina ficar sem stock de nenhum produto, altera o estado da máquina (MAQUINA.ESTADO\_MAQUINA) para 'SEM STOCK'.

**Abastecimento:** ao registar um abastecimento na tabela ABASTECE, a quantidade abastecida não pode ser negativa nem pode fazer com que o stock ultrapasse a capacidade do compartimento. Estas regras são validadas por uma trigger BEFORE INSERT. Depois do abastecimento, uma trigger AFTER INSERT atualiza o stock no compartimento, atualiza o campo QUANT\_EM\_MAQUINAS do produto e, caso aplicável, atualiza a DATA\_VISITA da máquina.

**Prevenção de abastecimentos repetidos:** se a mesma máquina for abastecida mais de uma vez no mesmo dia, isso pode gerar desperdício e custos desnecessários. Uma trigger BEFORE INSERT em ABASTECE verifica se já existe um abastecimento na mesma máquina (MAQUINA) no mesmo dia (via VISITA.DATA\_VISITA) e impede a operação com uma exceção.

**Operações dependentes do estado da viagem:** algumas operações (como abastecimento ou registo de visita) não devem ocorrer se a viagem associada ainda não começou. A data de início (VIAGEM.DATA\_INICIO) deve ser anterior à data atual. Isto deve ser validado por qualquer procedimento que execute ações dependentes de viagens.

**Monitorização de máquinas offline:** se uma máquina não reportar dados por mais de 30 minutos (verificado pelo campo MAQUINA.DATA\_ULTIMO\_REPORT), o seu estado deve ser atualizado para 'OFFLINE'. Isto pode ser feito por um procedimento agendado periodicamente (com DBMS\_SCHEDULER) que percorre as máquinas e verifica se o último report é demasiado antigo.

t)

## **As tabelas mais prováveis de ocuparem mais espaços são as tabelas:**

**VENDAS:** Pois cada venda gera um registo e com muitas máquinas a trabalhar diariamente esta tabela tende a crescer rapidamente.

**ABASTECE:** Cada reabastecimento gera uma linha normalmente são abastecidos vários produtos de uma vez.

**VISITA:** Regista todas as visitas sejam técnicas ou operacionais e cada reabastecimento ou intervenção gera uma visita.

**CONTÉM:** Relaciona cada viagem com os produtos transportados.

Normalmente cada linha é pequena, mas tem um grande volume por causa da quantidade de produtos.

**ENCOMENDA:** Gera registos sempre que há necessidade de repor o stock, esta cresce em previsão do consumo/vendas.

Esta estimativa será baseada no tamanho do bloco de dados do Oracle (4KB) e no parâmetro PCTFREE.

## **1- Tamanho médio de uma Linha:**

O tamanho de uma linha resulta da soma dos tamanhos dos seus atributos, mais um cabeçalho fixo de 5 bytes. Os tamanhos considerados foram:

- **Number :** 4 bytes;
- **Date :** 7 bytes;
- **Varchar2(n) :** n / 2 bytes (valor médio armazenado)

### a) Tabela VENDA

- ID\_VENDA (NUMBER): 4 bytes
- ID\_COMPARTIMENTO (NUMBER): 4 bytes
- ID\_PRODUTO (NUMBER): 4 bytes
- METODO\_PAGAMENTO (VARCHAR2(20), média de 10): 10 bytes
- DATA\_VENDA (DATE): 7 bytes
- **Total:** 5 (cabeçalho) + 4 + 4 + 4 + 10 + 7 = **34 bytes**

### b) Tabela ABASTECE

- ID\_COMPARTIMENTO (NUMBER): 4 bytes
- ID\_VISITA (NUMBER): 4 bytes
- ID\_PRODUTO (NUMBER): 4 bytes
- QUANT\_ABASTECIDA (NUMBER): 4 bytes
- **Total:** 5 + 4 + 4 + 4 + 4 = **21 bytes**

### c) Tabela VISITA

- ID\_VISITA (NUMBER): 4 bytes
- ID\_VIAGEM (NUMBER): 4 bytes
- ID\_MAQUINA (NUMBER): 4 bytes
- ESTADO\_MAQUINA (VARCHAR2(20), média 10): 10 bytes
- DATA\_VISITA (DATE): 7 bytes
- **Total:** 5 + 4 + 4 + 4 + 10 + 7 = **34 bytes**

### d) Tabela TRANSPORTA

- ID\_VIAGEM (NUMBER): 4 bytes
- ID\_PRODUTO (NUMBER): 4 bytes
- QUANTIDADE\_PRODUTO (NUMBER): 4 bytes
- **Total:** 5 + 4 + 4 + 4 = **17 bytes**

#### e) Tabela ENCOMENDA

- QUANT\_ENCOMENDA (NUMBER): 4 bytes
- ID\_ENCOMENDA (NUMBER): 4 bytes
- ARMAZEM\_ORIGEM (NUMBER): 4 bytes
- ARMAZEM\_DESTINO (NUMBER): 4 bytes
- ID\_ARMAZEM (NUMBER): 4 bytes
- ID\_PRODUTO (NUMBER): 4 bytes
- **Total:**  $5 + (6 \times 4) = 29 \text{ bytes}$

## 2- Registros por bloco

Considerando:

- Tamanho do bloco = 4096 bytes
- Overhead = 84 bytes
- Espaço útil = 4012 bytes
- PCTFREE = 0%

A partir do tamanho médio de cada linha calculado anteriormente, estimamos quantos registos cabem por bloco da seguinte forma:

- A tabela VENDA, com linhas de **34 bytes**, permite armazenar aproximadamente **118 registos por bloco (4012 dividido por 34)**.
- A tabela ABASTECE, com linhas de **21 bytes**, comporta cerca de **191 registos por bloco**.
- A tabela VISITA, também com linhas de **34 bytes**, armazena igualmente cerca de **118 registos por bloco**.
- A tabela TRANSPORTA, com linhas de **17 bytes**, permite inserir aproximadamente **236 registos por bloco**.
- Por fim, a tabela ENCOMENDA, com linhas de **29 bytes**, permite armazenar cerca de **138 registos por bloco**.

### 3- Estimativa de crescimento anual

Para estimar o número de regtos que cada tabela irá acumular num ano, consideraram-se cenários operacionais realistas baseados na utilização de 100 máquinas automáticas ao longo de 365 dias.

- A tabela **VENDA** representa todas as vendas realizadas. Se cada máquina efetuar em média **50 vendas por dia**, temos um total de **100 máquinas × 50 vendas/dia × 365 dias**, o que resulta em aproximadamente **1.825.000 regtos por ano**.
- A tabela **ABASTECE** regista os reabastecimentos de compartimentos. Supondo que cada máquina é reabastecida uma vez por semana, com cerca de **10 produtos diferentes**, temos: **100 máquinas × 1 reabastecimento/semana × 52 semanas × 10 produtos**, resultando em **52.000 regtos por ano**.
- A tabela **VISITA** corresponde às visitas efetuadas a cada máquina, normalmente para manutenção ou reabastecimento. Se cada máquina receber uma visita semanal, isso representa **100 máquinas × 52 semanas**, o que equivale a **5.200 regtos por ano**.
- A tabela **CONTEM** relaciona as viagens com os produtos transportados. Considerando que cada uma das **5.200 visitas** transporta, em média, **10 produtos diferentes**, temos **52.000 regtos por ano**.
- A tabela **ENCOMENDA** regista os pedidos de reposição feitos pelos armazéns. Estima-se que existam cerca de **1.000 encomendas** ao longo do ano, com uma média de **10 produtos por encomenda**, o que perfaz um total de **10.000 regtos por ano**.

## 4- Espaço anual estimado

Para calcular o espaço anual ocupado por cada tabela, divide-se o número de registos por ano pelo número de registos que cabem num bloco, obtendo-se assim o número de blocos necessários. Em seguida, multiplica-se o número de blocos pelo tamanho do bloco (4096 bytes) e converte-se o resultado para megabytes.

- A tabela **VENDA**, com **1.825.000 registos por ano** e **118 registos por bloco**, necessita de aproximadamente **15.466 blocos anuais**. Multiplicando esse valor por **4096 bytes** por bloco, obtemos cerca de **63,3 megabytes por ano**.
- A tabela **ABASTECE**, com **52.000 registos anuais** e **191 registos por bloco**, precisa de cerca de **273 blocos**. Isso corresponde a aproximadamente **1,1 megabytes de espaço por ano**.
- A tabela **VISITA**, com **5.200 registos por ano** e **118 registos por bloco**, requer cerca de **44 blocos**, ocupando cerca de **0,18 megabytes anuais**.
- A tabela **CONTEm**, com **52.000 registos por ano** e **236 registos por bloco**, necessita de cerca de **220 blocos**, o que equivale a aproximadamente **0,9 megabytes por ano**.
- Por fim, a tabela **ENCOMENDA**, com **10.000 registos por ano** e **138 registos por bloco**, consome cerca de **73 blocos**, o que representa aproximadamente **0,29 megabytes de espaço por ano**.