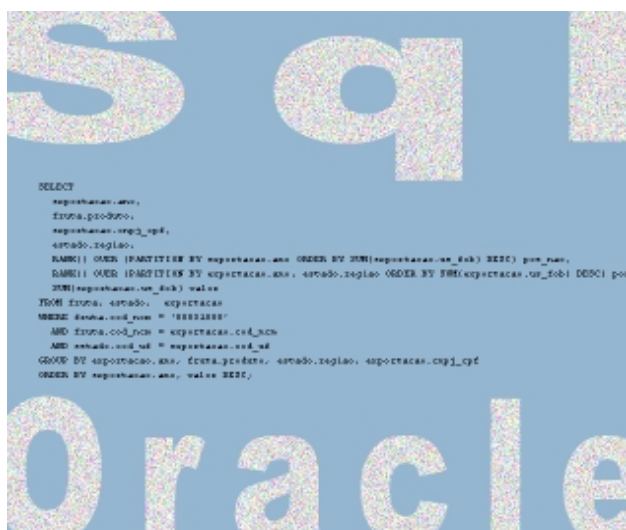


ISSN 1677-8464

Funções Analíticas da Linguagem Sql do Oracle

Carlos Alberto Alves Meira¹
Tatiana Aparecida Lima de Souza²



A linguagem *Structured Query Language* (SQL), até pouco tempo, tinha deficiência no suporte a tarefas de análise. Cálculos básicos de Inteligência de Negócios (*Business Intelligence*) ou *On-Line Analytical Processing* (OLAP), tais como médias móveis, classificações e comparações entre diferentes períodos, requeriam programação extensa em SQL ou em outra linguagem, apresentando problemas de desempenho (Oracle Corporation, 1999).

A Oracle introduziu funções analíticas na sua linguagem SQL com o sistema gerenciador de bancos de dados Oracle 8i. Depois disso, essas funções foram para revisão no instituto *American National Standards Institute* (ANSI), para inclusão no padrão SQL. Atualmente, as funções analíticas do Oracle fazem parte do padrão denominado SQL:1999, oficializadas no ano 2001, numa emenda à norma ANSI/ISO/IEC 9075 (ISO, 1992) que especifica o padrão.

As funções analíticas podem ser consideradas novas e existem poucos trabalhos que descrevem suas funcionalidades e exemplificam suas virtudes. Este trabalho é uma pequena, mas importante, contribuição nesse sentido, abordando algumas das funções analíticas do Oracle e exemplificando o seu uso, a partir de experiências obtidas num projeto de integração de informação (Garcia-Molina et al., 1999; Salgado &

Lóscio, 2001), sob responsabilidade da Embrapa Informática Agropecuária.

O referido projeto é a construção do Armazém de Dados da Fruticultura (ADF), um *data warehouse* que reúne informações de bancos de dados mantidos por diferentes instituições, a respeito de produção, comércio exterior e comercialização de frutas e alguns derivados. O objetivo desse banco de dados é tornar disponíveis os dados integrados para consultas, permitindo análises para suporte à tomada de decisões como, por exemplo, indicação e direcionamento de programas de incentivo à produção e à exportação de frutas no Brasil (Seixas Neto et al., 2000; Fileto et al., 2002).

O conteúdo deste trabalho inicia com uma descrição geral das funções analíticas. Depois, são descritas as características de cada função contemplada, com exemplos de sua utilização. Cada função apresentada foi bastante útil na elaboração dos relatórios de análise emitidos a partir do ADF (Meira et al., 2001). Os exemplos estão baseados no esquema de banco de dados desse *data warehouse*. Entretanto, vale ressaltar que esses exemplos e a apresentação de seus resultados foram elaborados para fins didáticos, não representando o formato dos relatórios destinados aos usuários finais do ADF.

¹ M.Sc. em Ciências de Computação e Matemática Computacional, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo, 13083-970 – Campinas, SP. (e-mail: carlos@cnptia.embrapa.br)

² Tecnóloga em Processamento de Dados, Bolsista do CNPq na Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo, 13083-970 – Campinas, SP. (e-mail: tatiana@cnptia.embrapa.br)

Para um melhor entendimento dos exemplos, é necessário que o leitor tenha conhecimento prévio da linguagem SQL (Date & Darwen, 1997; Kocchar et al., 2000), especificamente de seus principais comandos e cláusulas. O esquema de banco de dados do ADF não é apresentado neste trabalho, com o intuito de não estender muito o texto, o que não compromete a compreensão dos exemplos.

Funções Analíticas do Oracle

As funções analíticas são valiosas para todo tipo de processamento, desde suporte à decisão interativo até geração de relatórios em lote. Além disso, melhoram o desempenho de consultas ao banco de dados e a produtividade dos desenvolvedores.

As otimizações incorporadas com as funções analíticas melhoram significativamente o desempenho das consultas. As funções permitem que desenvolvedores implementem análises com código bem mais claro e conciso. Ações que antes requeriam múltiplos comandos SQL, ou o uso de linguagem procedimental, podem agora ser realizadas com comandos SQL simples. O novo código requerido é mais rápido de ser elaborado e mais fácil de ser mantido.

As funções analíticas do Oracle estão organizadas em quatro famílias de funções (Oracle Corporation, 1999). Neste trabalho, são apresentadas algumas das funções analíticas, aquelas de maior utilidade nos relatórios emitidos a partir do ADF, sendo descritas, incluindo exemplos, funções de três das famílias, a saber:

- *Ranking Family*: as funções dessa família permitem o cálculo da posição de uma linha do resultado de uma consulta em relação às demais linhas, baseado em um conjunto de medidas; suporta questões do tipo “apresente os vinte principais municípios produtores de laranja do Brasil”.
- *Reporting Aggregate Family*: essa família permite apresentação numa mesma linha de valores calculados em diferentes níveis de agregação³. Uma das tarefas analíticas mais comuns é a comparação de um valor não agregado com um valor agregado.
- *Lag/Lead Family*: essas funções permitem comparações entre valores de diferentes linhas de uma tabela, fundamentais no estudo de mudanças e variações na atividade de análise.

A quarta família, *Window Aggregate Family*, suporta questões do tipo “apresente um valor cumulativo de

exportações de frutas, mês a mês” e permite cálculos considerando períodos estipulados, como uma média móvel de 12 meses. Na verdade, não são funções novas incorporadas, mas palavras-chaves novas na sintaxe de funções já conhecidas, tais como AVG (média), SUM (soma), MIN (valor mínimo), MAX (valor máximo), COUNT (contagem de linhas) e outras.

As funções analíticas são o último conjunto de operações realizado numa consulta, exceto a cláusula final ORDER BY. Todas as junções e todas as cláusulas WHERE, GROUP BY e HAVING são completadas antes das funções analíticas serem processadas. Portanto, as funções analíticas podem aparecer apenas na lista do SELECT ou na cláusula final ORDER BY.

As funções analíticas permitem que o conjunto resultante da consulta seja dividido em grupos ordenados de linhas chamados de partições. Essa divisão pode estar baseada em qualquer coluna ou expressão. Um resultado de consulta pode ter uma única partição, com todas as linhas, ou várias pequenas partições com poucas linhas. Cada cálculo executado com uma função analítica está baseado numa linha corrente dentro de uma partição.

Função RANK

As funções da família *Ranking* calculam o *rank* (posição, ordem) de uma linha em relação às demais linhas, dentro de um conjunto de dados, baseado em um valor ou um conjunto de medidas definido pelo usuário.

A função RANK produz uma classificação ordenada de linhas começando com a posição 1. A sintaxe dela encontra-se a seguir, exceto os detalhes da cláusula ORDER BY, que é mais conhecida. A sintaxe completa dessa e das demais funções analíticas pode ser consultada no manual de referência da linguagem SQL do Oracle (Lorentz, 1999).

Sintaxe: *RANK()* OVER ([*query_partition_clause*] *order_by_clause*)

query_partition_clause: PARTITION BY *exp_list*

A cláusula opcional PARTITION BY é usada para definir onde o cálculo da posição é reinicializado. Caso nenhuma partição seja especificada, a classificação é feita em todo o conjunto resultante da consulta. A coluna cujos valores são classificados é definida na cláusula ORDER BY. A posição 1 é atribuída ao menor valor dessa coluna, exceto se uma ordenação descendente (DESC) for usada.

A consulta de exemplo a seguir produz como resultado uma classificação anual dos países importadores de castanha de caju (código de produto 8013200), de acordo com o valor exportado (US\$). Um trecho do resultado da consulta é apresentado na tabela seguinte, onde se pode perceber o reinício do ranking de uma partição (um determinado ano) para outra.

³ Agregação é o ato ou efeito de agregar (acumular ou juntar) valores. Diferentes níveis de agregação numa mesma dimensão acumulam mais ou menos valores. Por exemplo, o nível regional possui maior agregação que o nível estadual.

```

SELECT
    exportacao.ano,
    fruta.produto,
    pais.nome_pais,
    RANK() OVER (PARTITION BY ano ORDER BY SUM(exportacao.us_fob) DESC) posicao,
    SUM(exportacao.us_fob) valor
FROM pais, fruta, exportacao
WHERE fruta.cod_ncm = '8013200'
    AND fruta.cod_ncm = exportacao.cod_ncm
    AND pais.cod_pais = exportacao.cod_pais
GROUP BY exportacao.ano, pais.nome_pais, fruta.produto
ORDER BY exportacao.ano, valor DESC;

```

Ano	Produto	País	Posição	Valor
...
2001	Castanha de Caju	Angola	28	1.410
2001	Castanha de Caju	Finlândia	29	1.021
2002	Castanha de Caju	Estados Unidos	1	64.738.454
2002	Castanha de Caju	Canadá	2	6.652.375
2002	Castanha de Caju	Alemanha	3	2.170.695
2002	Castanha de Caju	Itália	4	2.070.205
2002	Castanha de Caju	França	5	1.965.655
2002	Castanha de Caju	Líbano	6	1.868.781
2002	Castanha de Caju	Portugal	7	961.575
...

A próxima consulta de exemplo produz como resultado dois *rankings* de empresas (código CNPJ ou CPF) exportadoras de laranja (código de produto 8013200), de acordo com o valor exportado (US\$). O primeiro *ranking* está particionado por ano; o segundo por ano

e região. Desta forma, uma certa empresa tem sua classificação nacional num determinado ano e neste mesmo ano tem sua classificação na região em que está localizada. Um trecho do resultado da consulta é apresentado na tabela seguinte.

```

SELECT
    exportacao.ano,
    fruta.produto,
    exportacao.cnpj_cpf,
    estado.regiao,
    RANK() OVER (PARTITION BY exportacao.ano ORDER BY SUM(exportacao.us_fob) DESC) pos_nac,
    RANK() OVER (PARTITION BY exportacao.ano, estado.regiao ORDER BY SUM(exportacao.us_fob) DESC) pos_reg,
    SUM(exportacao.us_fob) valor
FROM fruta, estado, exportacao
WHERE fruta.cod_ncm = '08051000'
    AND fruta.cod_ncm = exportacao.cod_ncm
    AND estado.cod_uf = exportacao.cod_uf
GROUP BY exportacao.ano, fruta.produto, estado.regiao, exportacao.cnpj_cpf
ORDER BY exportacao.ano, valor DESC;

```

Ano	Produto	Empresa CNPJ/CPF	Região	Posição Nacional	Posição Regional	Valor
...
2001	Laranja	16495483000182	NORDESTE	16	1	810
2001	Laranja	67626655000140	SUDESTE	16	14	810
2002	Laranja	52746419001243	SUDESTE	1	1	3.556.637
2002	Laranja	61649810000168	SUDESTE	2	2	2.550.461
2002	Laranja	50897875002150	SUDESTE	3	3	762.550
2002	Laranja	48155543000457	SUDESTE	4	4	279.419
2002	Laranja	65023467000100	SUDESTE	5	5	148.734
2002	Laranja	58902438000145	SUDESTE	6	6	49.233
2002	Laranja	53522801000183	SUDESTE	7	7	47.030
2002	Laranja	84430149000109	SUL	8	1	30.309
2002	Laranja	87178448000188	SUL	9	2	18.815
2002	Laranja	00026893000144	SUDESTE	10	8	8.564
...

Pelo resultado apresentado na tabela, verifica-se que, no ano 2002⁴: a empresa exportadora de laranja primeira colocada na região Sul ficou com a oitava posição nacional; as sete primeiras colocadas são da região Sudeste; a oitava colocada na região Sudeste ficou com a décima posição nacional.

Para gerar *rankings* com os N primeiros colocados, é preciso colocar a consulta que aplica a função analítica como uma subconsulta, e depois aplicar uma condição de filtro fora da subconsulta. A seleção dos N primeiros não funciona na consulta com a função analítica, pois, conforme descrito anteriormente, o cálculo dos valores resultantes da aplicação de funções analíticas é feito após o processamento da cláusula WHERE. O procedimento correto é ilustrado a seguir, supondo que se deseja selecionar as dez primeiras posições de um *ranking*:

```
SELECT * FROM
    (<consulta com a função analítica que
calcula rank>)
WHERE rank <= 10;
```

Função DENSE_RANK

A função DENSE_RANK é idêntica à função RANK, exceto em relação ao tratamento das ocorrências de valores repetidos. Nesse caso, a função DENSE_RANK não deixa "abertura" na seqüência numérica do ranking.

Sintaxe: *DENSE_RANK()* OVER
(*[query_partition_clause] order_by_clause*)

É mais fácil verificar a diferença entre as duas funções (RANK e DENSE_RANK) por meio de um exemplo. A consulta de exemplo a seguir produz como resultado a classificação dos municípios brasileiros produtores de abacate (código de produto 0401) em 2000, de acordo com a área colhida. Um trecho do resultado da consulta é apresentado na tabela seguinte.

⁴ Valores acumulados até outubro de 2002.

```

SELECT
  pam.ano,
  municipio.nome_municipio,
  estado.sigla_uf,
  pam.area_colhida,
  RANK() OVER (PARTITION BY pam.ano ORDER BY pam.area_colhida DESC) rank,
  DENSE_RANK() OVER (PARTITION BY pam.ano ORDER BY pam.area_colhida DESC) dense_rank
FROM pam, estado, municipio
WHERE municipio.cod_uf = pam.cod_uf
  AND municipio.cod_meso = pam.cod_meso
  AND municipio.cod_micro = pam.cod_micro
  AND municipio.cod_municipio = pam.cod_municipio
  AND municipio.cod_uf = estado.cod_uf
  AND pam.cod_produto = '0401'
  AND pam.ano = '2000'
GROUP BY pam.ano, municipio.nome_municipio, estado.sigla_uf, pam.area_colhida
ORDER BY pam.ano, pam.area_colhida DESC;

```

Ano	Município	UF	Area colhida	Rank	Dense_rank
2000	Presidente Olegário	MG	603	1	1
2000	Venda Nova do Imigrante	ES	400	2	2
2000	Santarém Novo	PA	350	3	3
2000	Arapongas	PR	300	4	4
2000	Moji-Mirim	SP	281	5	5
2000	Limeira	SP	259	6	6
2000	Jardinópolis	SP	244	7	7
2000	Cordeirópolis	SP	230	8	8
2000	Arealva	SP	209	9	9
2000	Bauru	SP	200	10	10
2000	Tupã	SP	200	10	10
2000	Conceição do Castelo	ES	200	10	10
2000	Duartina	SP	190	13	11
2000	Mirandópolis	SP	180	14	12
2000	Castelo	ES	150	15	13
...

Nota-se no resultado apresentado na tabela que existe um empate triplo na décima posição, onde a área colhida é 200 hectares. Pelo *ranking* com a função RANK, a próxima posição é a décima terceira, indicando o número de municípios apresentados até aquele

ponto. Já no *ranking* com a função DENSE_RANK, a próxima posição é a décima primeira, sem pular da posição 10 para a posição 13, indicando que ocorreram onze diferentes valores de área colhida até aquele mesmo ponto.

Função `RATIO_TO_REPORT`

A função `RATIO_TO_REPORT` pertence à família *Reporting Aggregate Family*. As funções dessa família permitem que uma linha de um resultado de consulta possua valores de diferentes níveis de agregação.

Comparar valores de um nível de agregação com valores de outro nível de agregação é essencial num trabalho analítico. Por exemplo, pode-se querer comparar a média de exportação de frutas dos municípios de um estado com a média estadual ou calcular o percentual de um total mensal em relação ao total do ano.

A função `RATIO_TO_REPORT` calcula a proporção de um valor em relação à agregação de um conjunto de valores. Ela pode ser usada para calcular frações, tais como participações de mercado de empresas concorrentes e distribuição de custos em análises financeiras.

Sintaxe: `RATIO_TO_REPORT(expr) OVER`

([query_partition_clause])

A consulta de exemplo a seguir produz como resultado a relação de países importadores de laranja (código de

produto 08051000) do Brasil, indicando o total anual importado em valores monetários (US\$) e o percentual desse valor em relação ao total geral exportado de laranja pelo Brasil no mesmo ano. Um trecho do resultado (ano 2002 completo) da consulta é

```
SELECT
    exportacao.ano,
    fruta.produto,
    pais.nome_pais,
    SUM(exportacao.us_fob) valor,
    ROUND((RATIO_TO_REPORT(SUM(exportacao.us_fob))
OVER (PARTITION BY exportacao.ano))*100, 3) AS
percentual
FROM exportacao, fruta, pais
WHERE fruta.cod_ncm = '08051000'
    AND fruta.cod_ncm = exportacao.cod_ncm
    AND pais.cod_pais = exportacao.cod_pais
GROUP BY exportacao.ano, fruta.produto,
pais.nome_pais
ORDER BY exportacao.ano, valor DESC;
```

	<i>Produto</i>	<i>País</i>	<i>Valor</i>	<i>Percentual</i>
...
2002	Laranja	Países Baixos	3.104.024	41,581
2002	Laranja	Espanha	1.876.054	25,131
2002	Laranja	Emir. Árabes Un.	838.710	11,235
2002	Laranja	Reino Unido	722.106	9,673
2002	Laranja	Portugal	512.580	6,866
2002	Laranja	Coveite	236.414	3,167
2002	Laranja	Indonésia	97.516	1,306
2002	Laranja	Arábia Saudita	48.510	0,650
2002	Laranja	França	18.815	0,252
2002	Laranja	Cabo Verde	5.706	0,076
2002	Laranja	Angola	3.098	0,042
2002	Laranja	Uruguai	1.248	0,017
2002	Laranja	Chile	183	0,002
2002	Laranja	Argentina	78	0,001

apresentado na tabela seguinte.

O valor total exportado de laranja em 2002⁴ pelo Brasil foi de US\$ 7.465.042,00 (somatório dos valores da coluna VALOR). Para exemplificar como é feito o cálculo do percentual de um país em relação a esse total anual geral, considere as exportações de laranja para os Países Baixos. O cálculo do percentual é feito da seguinte forma (a função ROUND arredonda o resultado da expressão para o número de casas decimais especificado após a vírgula, no caso 3):

Percentual = ROUND((3.104.024 / 7.465.042)*100, 3) = 41,581

A fração 3.104.024 (total de Países Baixos em 2002) dividido por 7.465.042 (total anual geral) é calculada pela função RATIO_TO_REPORT (ver consulta de exemplo anterior).

Funções LAG e LEAD

Uma tarefa primordial em qualquer processo de análise é a comparação de valores dentro de conjuntos de dados. Pode-se querer analisar mudanças nas vendas mensais do ano corrente comparadas com as de anos anteriores ou analisar a variação entre orçamento e custos reais, por exemplo.

As funções LAG e LEAD permitem comparações entre duas linhas de um mesmo conjunto de dados

```
SELECT
    exportacao.ano,
    fruta.produto,
    SUM(exportacao.us_fob) valor,
    LAG (SUM(exportacao.us_fob), 1) OVER (ORDER BY exportacao.ano) anterior,
    LEAD (SUM(exportacao.us_fob), 1) OVER (ORDER BY exportacao.ano) posterior
FROM fruta, exportacao
WHERE fruta.cod_ncm = exportacao.cod_ncm
    AND fruta.cod_ncm = '08051000'
GROUP BY exportacao.ano, fruta.produto
ORDER BY exportacao.ano;
```

especificando um deslocamento a partir da linha corrente, sem a necessidade de realizar *self-joins* (junção de uma tabela com ela mesma), o que melhora sensivelmente o desempenho da consulta.

A função LAG realiza o deslocamento no sentido inverso da ordenação do conjunto de dados, enquanto a função LEAD realiza o deslocamento no mesmo sentido da ordenação. A sintaxe dessas funções encontra-se a seguir. *Offset* é o deslocamento, que assume valor 1 caso não seja especificado. *Default* é um valor opcional de deslocamento, caso o *offset* especificado ultrapasse o limite do conjunto de dados.

Sintaxe: {LAG | LEAD} (value_expr [, offset] [, default])
OVER ([query_partition_clause] order_by_clause)

A consulta de exemplo a seguir produz como resultado, em cada linha e na coluna VALOR, o total anual de laranja exportado pelo Brasil⁴ (código de produto 08051000) em valores monetários (US\$). A coluna ANTERIOR indica o total anual exportado no ano anterior (offset = 1), calculado com a função LAG. E a coluna POSTERIOR indica o total anual exportado no ano seguinte, calculado com a função LEAD. O resultado completo da consulta é apresentado na tabela seguinte.

Ano	Fruta	Valor	Anterior	Posterior
1997	Laranja	23.091.638	NULL	14.358.728
1998	Laranja	14.358.728	23.091.638	21.108.431
1999	Laranja	21.108.431	14.358.728	15.247.625
2000	Laranja	15.247.625	21.108.431	27.538.089
2001	Laranja	27.538.089	15.247.625	6.360.426
2002	Laranja	6.360.426	27.538.089	NULL

O primeiro valor na coluna ANTERIOR e o último valor na coluna POSTERIOR são nulos, pois não existem registros para os anos 1996 e 2003, respectivamente. Vale ressaltar também que não foi necessária a cláusula PARTITION BY, já que as funções analíticas deviam ser aplicadas no resultado completo da consulta.

O exemplo apresentado é bastante simples, apenas para finalidade didática. É claro que é mais interessante para o usuário final analisar as variações já calculadas, sejam absolutas ou relativas. Esses cálculos, a partir dos valores apresentados, são fáceis de serem feitos utilizando operações simples de subtração e divisão.

Considerações Finais

Este trabalho apresentou as funções analíticas da linguagem SQL do sistema gerenciador de bancos de dados Oracle, que também fazem parte do padrão ANSI/ISO/IEC SQL:1999. Portanto, tem utilidade não apenas para desenvolvedores Oracle, mas também para qualquer desenvolvedor que utilize um produto com linguagem SQL compatível com o referido padrão.

Não se fez uma descrição completa e exaustiva das funções analíticas. Pelo contrário, procurou-se fazer uma breve apresentação dessas funções, descrever sucintamente algumas delas, mas principalmente, apresentar exemplos didáticos de sua utilidade. É importante salientar que as demais funções possuem sintaxe e processamento semelhantes. Sendo assim, após ler este trabalho, o leitor terá facilidade para ler e compreender os documentos técnicos de referência citados no texto.

Apenas para informação, as demais funções analíticas são: CUME_DIST e PERCENT_RANK, para cálculo de distribuição cumulativa; NTILE, para cálculo de quartis e outros Ntis; e ROW_NUMBER, para numeração de linhas com base na cláusula ORDER_BY da consulta.

As funções analíticas ganharam importância com o grande aumento da demanda por sistemas de integração de dados (*data warehouses*) e sistemas de suporte a decisão. Entretanto, qualquer sistema de informação pode se beneficiar das suas funcionalidades, pois, como visto, suas operações de análise são comuns a muitos tipos de aplicações.

Referências Bibliográficas

DATE, C. J.; DARWEN, H. **Guide to SQL standard: user's guide to the standard database language SQL**. 4.ed. Reading: Addison-Wesley, 1997. 522 p.

FILETO, R.; MEIRA, C. A. A.; NAKA, J.; SEIXAS NETO, A.; MEDEIROS, C. B. An XML - Centered warehouse to manage information of the fruit supply chain. In: WORLD CONGRESS OF COMPUTER IN AGRICULTURE AND NATURAL RESOURCES, 2002, Iguazu Falls. **Proceedings of the World Congress of Computers in Agriculture and Natural Resources**. Iguazu Falls: ASAE, 2001. p. 540-547.

GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOW, J. Information integration. In: GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOW, J. **Database system implementation**. New York: Prentice Hall, 1999. Ch.11, p. 595-641.

ISO. **International standard ISO/IEC 9075: information technology - database languages - SQL**. 3rd ed. Geneve, 1992.

KOCHHAR, N.; GRAVINA, E.; NATHAN, P. **Introdução ao Oracle: SQL e PL/SQL: guia do estudante**. Redwood Shores: Oracle Corporation, 2000. Paginação irregular. (Oracle University).

LORENTZ, D. **Oracle8i: SQL reference, release 2 (8.1.6)**. Redwood City: Oracle Corporation, 1999. Paginação irregular.

MEIRA, C. A. A.; SEIXAS NETO, A.; NAKA, J.; CUNHA, L. M. S. **Análise de comércio exterior de frutas a partir do armazém de dados da fruticultura brasileira**. Campinas: Embrapa Informática Agropecuária, 2001. 6 p. (Embrapa Informática Agropecuária. Comunicado Técnico, 17).

ORACLE CORPORATION. **Analytic functions for Oracle8i: an Oracle technical white paper**. Redwood Shores, 1999. 20 p.

SALGADO, A. C.; LÓSCIO, B. F. Integração de dados na Web. In: ESCOLA REGIONAL DE INFORMÁTICA DA SBC - REGIONAL DE SÃO PAULO, 6., 2001, São Carlos. **Minicursos: coletânea de textos: anais**. São Carlos: USP-ICMC, 2001. p. 157-174.

SEIXAS NETO, A.; CUNHA, L. M. S.; MEIRA, C. A. A. **Sistema de integração e qualificação de informação para a cadeia de frutas**. Campinas: Embrapa Informática Agropecuária, 2000. 19 p. (Embrapa. Programa 14 - Intercâmbio e Produção de Informação em Apoio às Ações de Pesquisa e Desenvolvimento. Projeto 14.2001.368). Projeto em andamento.

Comunicado Técnico, 41

**Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)**
Av. André Tosello, 209
Cidade Universitária - "Zeferino Vaz"
Barão Geraldo - Caixa Postal 6041
13083-970 - Campinas, SP
Telefone (19) 3789-5743 - Fax (19) 3289-9594
e-mail: sac@cnptia.embrapa.br

1ª edição
2002 - on-line
Todos os direitos reservados

Comitê de Publicações

Presidente: José Ruy Porto de Carvalho
Membros efetivos: Amarindo Fausto Soares, Ivanilde Dispatto, Luciana Alvim Santos Romani, Marcia Izabel Fugisawa Souza, Suzilei Almeida Carneiro
Suplentes: Adriana Delfino dos Santos, Fábio Cesar da Silva, João Francisco Gonçalves Antunes, Maria Angélica de Andrade Leite, Moacir Pedroso Júnior

Expediente

Supervisor editorial: Ivanilde Dispatto
Normalização bibliográfica: Marcia Izabel Fugisawa Souza
Capa: Intermídia Publicações Científicas
Editoração Eletrônica: Intermídia Publicações Científicas