

Sistemas de Informação

Design e Desenvolvimento de Banco de Dados

PROF. MILTON

Agregando Datos Usando Funciones de Grupo

Objetivos

- Depois de completar esta lição, você poderá fazer o seguinte:
 - Identificar as funções de grupo disponíveis
 - Descrever o uso de funções de grupo
 - Agrupar dados usando a cláusula GROUP BY
 - Incluir ou excluir linhas agrupadas usando a cláusula HAVING

Objetivo da Lição

Esta lição aborda funções. Visa obter informações sumariadas, tais como médias, para grupos de linhas. Discute como agrupar linhas de uma tabela em conjuntos menores e como especificar critérios de pesquisa para grupos de linhas.

O Que São Funções de Grupo?

- As funções de grupo operam em conjuntos de linhas para fornecer um resultado por grupo.

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

"salário
máximo na
tabela EMP"

MAX (SAL)

5000

Funções de Grupo

De modo diferente das funções de uma única linha, as **funções de grupo** operam em conjuntos de linhas para fornecer um resultado por grupo. Esses conjuntos podem ser a tabela inteira ou a tabela dividida em grupos.

Tipos de Funções de Grupo

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE

Funções de Grupo (continuação)

Cada uma das funções aceita um argumento. A tabela a seguir identifica as opções que podem ser usadas na sintaxe:

Função	Descrição
AVG([DISTINCT <u>ALL</u>] <i>n</i>)	Valor médio de <i>n</i> , ignorando valores nulos
COUNT({*[DISTINCT <u>ALL</u>] <i>expr</i> })	Número de linhas, onde <i>expr</i> avalia para algo diferente de nulo (Conte todas as linhas selecionadas usando *, inclusive duplicadas e linhas com nulos.)
MAX([DISTINCT <u>ALL</u>] <i>expr</i>)	Valor máximo de <i>expr</i> , ignorando valores nulos
MIN([DISTINCT <u>ALL</u>] <i>expr</i>)	Valor mínimo de <i>expr</i> , ignorando valores nulos
STDDEV([DISTINCT <u>ALL</u>] <i>x</i>)	Desvio padrão de <i>n</i> , ignorando valores nulos
SUM([DISTINCT <u>ALL</u>] <i>n</i>)	Valores somados de <i>n</i> , ignorando valores nulos
VARIANCE([DISTINCT <u>ALL</u>] <i>x</i>)	Variação de <i>n</i> , ignorando valores nulos

Usando Funções de Grupo

```
SELECT      [coluna,] group_function(coluna)
FROM        tabela
[WHERE      condição]
[GROUP BY   coluna]
[ORDER BY   coluna];
```

Diretrizes para o Uso de Funções de Grupo

DISTINCT faz com que a função considere somente valores não-duplicados; ALL faz com que ela considere cada valor, inclusive duplicados. O default é ALL e, portanto, não precisa ser especificado.

Os tipos de dados para os argumentos podem ser CHAR, VARCHAR2, NUMBER ou DATE, onde *expr* está listado.

Todas as funções de grupo, exceto COUNT(*), ignoram valores nulos. Para substituir um valor por valores nulos, use a função NVL.

O Oracle Server classifica implicitamente a definição do resultado em ordem crescente quando usa uma cláusula GROUP BY. Para sobrepor essa ordenação default, DESC pode ser usado em uma cláusula ORDER BY.

Usando Funções AVG e SUM

- Você pode usar AVG e SUM para dados numéricos.

```
SQL> SELECT AVG(sal), MAX(sal),  
2          MIN(sal), SUM(sal)  
3 FROM emp  
4 WHERE job LIKE 'SALES%';
```

AVG (SAL)	MAX (SAL)	MIN (SAL)	SUM (SAL)	
-----	-----	-----	-----	
1400	1600	1250	5600	

Funções de Grupo

Você pode usar as funções **AVG**, **SUM**, **MIN** e **MAX** com colunas que possam armazenar dados numéricos. O exemplo no slide exibe os salários maior, médio, menor e a soma dos salários mensais de todos os vendedores.

Usando Funções MIN e MAX

- Você pode usar MIN e MAX para qualquer tipo de dados.

```
SQL> SELECT MIN(hiredate), MAX(hiredate)
2 FROM emp;
```

MIN (HIRED)	MAX (HIRED)
-----	-----
17-DEC-80	12-JAN-83

Funções de Grupo (continuação)

Você pode usar as funções **MAX** e **MIN** para qualquer tipo de dados. O exemplo no slide exibe o funcionário mais antigo e o mais novo.

O exemplo a seguir exibe o primeiro e o último nome de funcionário em uma lista alfabética de todos os funcionários.

```
SQL> SELECT MIN(ename), MAX(ename)
2 FROM emp;
```

MIN (ENAME)	MAX (ENAME)
-----	-----
ADAMS	WARD

Observação: As funções AVG, SUM, VARIANCE e STDDEV só podem ser usadas com tipos de dados numéricos.

Usando a Função COUNT

- COUNT(*) retorna o número de linhas em uma tabela.

```
SQL> SELECT COUNT (*)  
2 FROM emp  
3 WHERE deptno = 30;
```

```
COUNT (*)  
-----  
6
```

A Função COUNT

A Função **COUNT** tem dois formatos:

COUNT(*)

COUNT(*expr*)

COUNT(*) retorna o número de linhas em uma tabela, inclusive linhas duplicadas e linhas contendo valores nulos em qualquer uma das colunas. Se uma cláusula WHERE estiver incluída na instrução SELECT, COUNT(*) retornará o número de linhas que satisfizer a condição na cláusula WHERE.

Entretanto, COUNT(*expr*) retorna o número de linhas não nulas na coluna identificada por *expr*.

O exemplo no slide exibe o número de funcionários no departamento 30.

Usando a Função COUNT

- COUNT(*expr*) retorna o número de linhas não nulas.

```
SQL> SELECT COUNT(comm)
2 FROM emp
3 WHERE deptno = 30;
```

```
COUNT (COMM)
-----
4
```

10

A Função COUNT (continuação)

O exemplo no slide exibe o número de funcionários no departamento 30 que podem receber uma comissão. Observe que o resultado fornece o número total de quatro linhas porque dois funcionários no departamento 30 não podem receber a comissão e, portanto, a coluna COMM contém um valor nulo.

Exemplo

Exiba o número de departamentos na tabela EMP.

```
SQL> SELECT COUNT(deptno)
2 FROM emp;
```

```
COUNT (DEPTNO)
-----
14
```

Exiba o número de departamentos distintos na tabela EMP.

```
SQL> SELECT COUNT(DISTINCT (deptno))
2 FROM emp;
```

```
COUNT (DISTINCT (DEPTNO) )
-----
3
```

Funções de Grupo e Valores Nulos

- As funções de grupo ignoram valores nulos na coluna.

```
SQL> SELECT AVG(comm)
      2 FROM emp;
```

```
AVG (COMM)
-----
      550
```

Funções de Grupo e Valores Nulos

Todas as funções de grupo, com a exceção de COUNT (*), ignoram valores nulos na coluna. No exemplo do slide, a média é calculada com base *somente* nas linhas da tabela em que um valor válido é armazenado na coluna COMM. A média é calculada como o total da comissão sendo paga a todos os funcionários dividido pelo número de funcionários recebendo a comissão (4).

Usando a Função NVL com Funções de Grupo

- A função NVL força as funções de grupo a incluírem valores nulos.

```
SQL> SELECT AVG (NVL (comm, 0) )  
2 FROM emp;
```

```
AVG (NVL (COMM, 0) )  
-----  
157.14286
```

Funções de Grupo e Valores Nulos (continuação)

A função NVL força as funções de grupo a incluírem valores nulos. No exemplo do slide, a média é calculada com base em *todas* as linhas na tabela, independentemente de os valores nulos estarem armazenados na coluna COMM. A média é calculada como o total da comissão sendo paga a todos os funcionários dividido pelo número total de funcionários na empresa (14).

Criando Grupos de Dados

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

2916.6667

2175

1566.6667

"salário
médio
na tabela
EMP
para cada
departamento
"

DEPTNO	AVG (SAL)
10	2916.6667
20	2175
30	1566.6667

Grupos de Dados

Até este momento, todas as funções de grupo trataram a tabela como um grande grupo de informações. Às vezes, é necessário dividir a tabela de informações em grupos menores. Isso pode ser feito usando a cláusula **GROUP BY**.

Criando Grupos de Dados: Cláusula GROUP BY

```
SELECT      coluna, group_function(coluna)
FROM        tabela
[WHERE      condição]
[GROUP BY   group_by_expression]
[ORDER BY   coluna];
```

- Divida linhas de uma tabela em grupos menores usando a cláusula GROUP BY.

A Cláusula GROUP BY

Você pode usar a cláusula **GROUP BY** para dividir as linhas de uma tabela em grupos. Em seguida, pode usar as funções de grupo para retornar informações sumárias para cada grupo.

Na sintaxe:

group_by_expression especifica colunas cujos valores determinam a base para

agrupar linhas

Diretrizes

Se você incluir uma função de grupo em uma cláusula SELECT, não poderá selecionar resultados individuais, *a menos que* a coluna individual apareça na cláusula GROUP BY. Se você não conseguir incluir a lista de colunas, uma mensagem de erro será exibida.

Ao usar uma cláusula WHERE, você pode excluir linhas com antecedência antes de dividi-las em grupos.

Você deve incluir as *colunas* na cláusula GROUP BY.

Não é possível usar o apelido de coluna na cláusula GROUP BY.

Por default, as linhas são classificadas por ordem crescente das colunas incluídas na lista GROUP BY. Isso pode ser sobreposto usando a cláusula ORDER BY.

Usando a Cláusula GROUP BY

- Todas as colunas na lista SELECT que não estejam em funções de grupo devem estar na cláusula GROUP BY.

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno;
```

DEPTNO	AVG(SAL)
10	2916.6667
20	2175
30	1566.6667

A Cláusula GROUP BY (continuação)

Quando utilizar a cláusula GROUP BY, certifique-se de que todas as colunas na lista SELECT que não estejam nas funções de grupo estejam incluídas na cláusula GROUP BY. O exemplo no slide exibe o número do departamento e o salário médio para cada departamento. Essa instrução SELECT, que contém uma cláusula GROUP BY, é avaliada da seguinte forma:

A cláusula SELECT especifica as colunas a serem recuperadas:

A coluna de números de departamento na tabela EMP

A média de todos os salários no grupo que você especificou na cláusula GROUP BY

A cláusula FROM especifica as tabelas que o banco de dados deve acessar:
a tabela EMP.

A cláusula WHERE especifica as linhas a serem recuperadas. Já que não há uma cláusula WHERE, todas as linhas são recuperadas por default.

A cláusula GROUP BY especifica como as linhas devem ser agrupadas. As linhas são agrupadas pelo número do departamento, de forma que a função AVG que esteja sendo aplicada à coluna de salários calcule o *salário médio para cada departamento*.

Usando a Cláusula GROUP BY

- A coluna GROUP BY não precisa estar na lista SELECT.

```
SQL> SELECT      AVG(sal)
  2  FROM        emp
  3  GROUP BY deptno;
```

```
AVG(SAL)
-----
2916.6667
2175
1566.6667
```

A Cláusula GROUP BY (continuação)

A coluna GROUP BY não precisa estar na cláusula SELECT. Por exemplo, a instrução SELECT no slide exibe os salários médios para cada departamento sem exibir os respectivos números dos departamentos. Sem os números dos departamentos, entretanto, os resultados não parecem significativos.

Você pode usar a função de grupo na cláusula ORDER BY.

```
SQL> SELECT      deptno, AVG(sal)
  2  FROM        emp
  3  GROUP BY    deptno
  4  ORDER BY    AVG(sal);
```

```
DEPTNO      AVG(SAL)
-----
30          1566.6667
20              2175
10          2916.6667
```


Agrupando por Mais de Uma Coluna

EMP

DEPTNO	JOB	SAL
10	MANAGER	2450
10	PRESIDENT	5000
10	CLERK	1300
20	CLERK	800
20	CLERK	1100
20	ANALYST	3000
20	ANALYST	3000
20	MANAGER	2975
30	SALESMAN	1600
30	MANAGER	2850
30	SALESMAN	1250
30	CLERK	950
30	SALESMAN	1500
30	SALESMAN	1250

"soma de salários na tabela EMP para cada cargo, agrupados por departamento"

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600

Grupos Dentro de Grupos

Às vezes há necessidade de ver os resultados para grupos dentro de grupos. O slide mostra um relatório que exibe o salário total sendo pago a cada cargo, dentro de cada departamento.

A tabela EMP é agrupada primeiro pelo número do departamento e, dentro desse agrupamento, é agrupada pelo cargo. Por exemplo, os dois escriturários no departamento 20 estão agrupados e um único resultado (salário total) é produzido para todos os vendedores dentro do grupo.

Usando a Cláusula GROUP BY em Várias Colunas

```
SQL> SELECT deptno, job, sum(sal)
2 FROM emp
3 GROUP BY deptno, job;
```

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
...		

9 rows selected.

Grupos Dentro de Grupos (continuação)

É possível retornar resultados sumários para grupos e subgrupos listando mais de uma coluna GROUP BY. Você pode determinar a ordem de classificação default dos resultados pela ordem das colunas na cláusula GROUP BY. A instrução SELECT no slide, que contém uma cláusula GROUP BY, é avaliada da seguinte forma:

A cláusula SELECT especifica a coluna a ser recuperada:

- O número do departamento na tabela EMP

- O cargo na tabela EMP

- A soma de todos os salários no grupo que você especificou na cláusula GROUP BY

A cláusula FROM especifica as tabelas que o banco de dados deve acessar: a tabela EMP.

A cláusula GROUP BY especifica como você deve agrupar as linhas:

- Primeiro, as linhas são agrupadas pelo número do departamento

- Em seguida, dentro dos grupos de números de departamentos, as linhas são agrupadas pelo cargo

Dessa forma, a função SUM é aplicada à coluna de salários para todos os cargos dentro de cada grupo de números de departamentos.

Consultas Ilegais Usando Funções de Grupo

- Qualquer coluna ou expressão na lista SELECT que não seja uma função agregada deve estar na cláusula GROUP BY.

```
SQL> SELECT deptno, COUNT(ename)
2 FROM emp;
```

Coluna ausente na cláusula GROUP BY

```
SELECT deptno, COUNT(ename)
*
ERROR at line 1:
ORA-00937: Nenhuma função de grupo de grupo único
(Not a single-group group function)
```

19

Consultas Ilegais Usando Funções de Grupo

Sempre que você usar uma mistura de itens individuais (DEPTNO) e funções de grupo (COUNT) na mesma instrução SELECT, deve incluir uma cláusula GROUP BY que especifique os itens individuais (neste caso, DEPTNO). Se a cláusula GROUP BY estiver ausente, aparecerá a mensagem de erro "Nenhuma função de grupo de grupo único" e um asterisco (*) apontará para a coluna que contiver o erro. Você pode corrigir o

erro no slide ao adicionar a cláusula GROUP BY.

```
SQL> SELECT deptno, COUNT(ename)
2 FROM emp
3 GROUP BY deptno;

DEPTNO COUNT(ENAME)
```

```
-----
10      3
20      5
30      6
```

Qualquer coluna ou expressão na lista SELECT que não seja uma função agregada deve estar na cláusula GROUP BY.

Consultas Ilegais

Usando Funções de Grupo

- Não é possível usar a cláusula WHERE para restringir grupos.
- Use a cláusula HAVING para restringir grupos.

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 WHERE AVG(sal) > 2000
4 GROUP BY deptno;
```

```
WHERE AVG(sal) > 2000
*
ERROR at line 3:
ORA-00934: A função de grupo não é permitida aqui
(Group function is not allowed here)
```



Consultas Ilegais Usando Funções de Grupo (continuação)

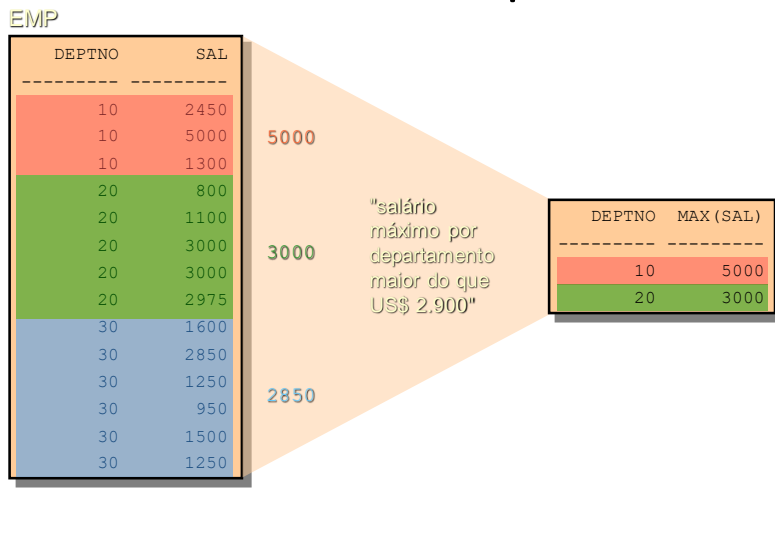
A cláusula WHERE não pode ser usada para restringir grupos. A instrução SELECT no slide resulta em um erro porque usa a cláusula WHERE para restringir a exibição de salários médios dos departamentos que tenham um salário médio maior do que US\$ 2.000.

Você pode corrigir o erro no slide usando a cláusula HAVING para restringir grupos.

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno
4 HAVING AVG(sal) > 2000;
```

DEPTNO	AVG(SAL)
10	2916.6667
20	2175

Excluindo Resultados do Grupo



Restringindo Resultados do Grupo

Da mesma forma que você usa a cláusula WHERE para restringir as linhas que seleciona, pode usar a cláusula **HAVING** para restringir grupos. Para localizar o salário máximo de cada departamento, mas mostrar apenas os departamentos que tenham um salário máximo de mais do que US\$ 2.900, faça o seguinte:

Localize o salário médio para cada departamento ao agrupar por número do departamento.

Restrinja os grupos aos departamentos com um salário máximo maior do que US\$ 2.900.

Excluindo Resultados do Grupo: Cláusula HAVING

- Use a cláusula HAVING para restringir grupos
 - As linhas são agrupadas.
 - A função de grupo é aplicada.
 - Os grupos que correspondem à cláusula HAVING são exibidos.

```
SELECT      coluna, group_function
FROM        tabela
[WHERE      condição]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   coluna];
```



A Cláusula HAVING

Use a cláusula **HAVING** para especificar quais grupos serão exibidos. Portanto, restrinja ainda mais os grupos com base nas informações agregadas.

Na sintaxe:

group_condition restringe os grupos de linhas
retornados aos grupos para os quais
a condição

especificada seja TRUE

O Oracle Server executa as seguintes etapas quando você usa a cláusula HAVING:

As linhas são agrupadas.

A função de grupo é aplicada ao grupo.

Os grupos que correspondem aos critérios na cláusula HAVING são exibidos.

A cláusula HAVING pode preceder a cláusula GROUP BY, mas recomenda-se que você coloque a cláusula GROUP BY primeiro, por ser mais lógico. Os grupos são formados e as funções de grupo são calculadas antes de a cláusula HAVING ser aplicada aos grupos na lista SELECT.

Usando a Cláusula HAVING

```
SQL> SELECT deptno, max(sal)
2 FROM emp
3 GROUP BY deptno
4 HAVING max(sal) > 2900;
```

DEPTNO	MAX (SAL)
10	5000
20	3000

A Cláusula HAVING (continuação)

O exemplo no slide exibe números de departamentos e o salário máximo para os departamentos cujo salário máximo seja maior do que US\$ 2.900.

Você pode usar a cláusula GROUP BY sem usar uma função de grupo na lista SELECT. Se você restringir linhas baseado no resultado de uma função de grupo, deve ter uma cláusula GROUP BY assim como a cláusula HAVING.

O exemplo a seguir exibe os números de departamentos e o salário médio para os departamentos cujo salário máximo seja maior do que US\$ 2.900:

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno
4 HAVING MAX(sal) > 2900;
```

DEPTNO	AVG (SAL)
10	2916.6667
20	2175

Usando a Cláusula HAVING

```
SQL> SELECT      job, SUM(sal) PAYROLL
  2  FROM          emp
  3  WHERE         job NOT LIKE 'SALES%'
  4  GROUP BY     job
  5  HAVING        SUM(sal)>5000
  6  ORDER BY     SUM(sal);
```

JOB	PAYROLL
ANALYST	6000
MANAGER	8275

A Cláusula HAVING (continuação)

O exemplo no slide exibe o cargo e o salário mensal total para cada cargo com uma folha de pagamento total excedendo US\$ 5.000. O exemplo exclui vendedores e classifica a lista pelo salário mensal total.

Aninhando Funções de Grupo

- Exiba o salário médio máximo.

```
SQL> SELECT max (avg (sal) )  
2 FROM emp  
3 GROUP BY deptno;
```

```
MAX (AVG (SAL) )  
-----  
2916.6667
```

Aninhando Funções de Grupo

As funções de grupo podem ser aninhadas até uma profundidade de dois. O exemplo no slide exibe o salário médio máximo.

Sumário

```
SELECT      coluna, group_function(coluna)
FROM        tabela
[WHERE      condição]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   coluna];
```

- Ordem de avaliação das cláusulas:
 - cláusula WHERE
 - cláusula GROUP BY
 - cláusula HAVING

Sumário

Sete funções de grupo estão disponíveis no SQL:

AVG
COUNT
MAX
MIN
SUM
STDDEV
VARIANCE

Você pode criar subgrupos usando a cláusula GROUP BY. Os grupos podem ser excluídos usando a cláusula HAVING.

Coloque as cláusulas HAVING e GROUP BY após a cláusula WHERE em uma instrução.

Coloque a cláusula ORDER BY por último.

O Oracle Server avalia as cláusulas na seguinte ordem:

Se a instrução contém uma cláusula WHERE, o servidor estabelece as linhas do candidato.

O servidor identifica os grupos especificados na cláusula GROUP BY.

A cláusula HAVING restringe ainda mais os grupos de resultado que não atendam aos critérios do grupo na cláusula HAVING.

Visão Geral do Exercício

- Mostrando diferentes consultas que usam funções de grupo
- Agrupando por linhas para obter mais de um resultado
- Excluindo grupos usando a cláusula HAVING

Visão Geral do Exercício

Ao final desse exercício, você deverá estar familiarizado com o uso de funções de grupo e a seleção de grupos de dados.

Questões Impressas

No caso das questões 1–3, marque Verdadeiro ou Falso.

Observação: Os apelidos de colunas são usados para as consultas.

Exercício 13

Determine a validade das afirmações a seguir. Marque Verdadeiro ou Falso.

1.As funções de grupo operam em muitas linhas para produzir um resultado por grupo.

Verdadeiro/Falso

2.As funções de grupo incluem nulos nos cálculos.

Verdadeiro/Falso

3.A cláusula WHERE restringe as linhas antes da inclusão em um cálculo de grupo.

Verdadeiro/Falso

4.Exiba os salários maior, médio, menor e a soma de todos os salários de todos os funcionários. Coloque um label nas colunas Maximum, Minimum, Sum e Average, respectivamente. Arredonde os resultados para o número inteiro mais próximo.

Maximum	Minimum	Sum	Average
-----	-----	-----	-----
5000	800	29025	2073

5.Modifique o exercício anterior para exibir o salário maior, médio, menor e a soma de todos os salários para cada tipo de cargo.

JOB	Maximum	Minimum	Sum	Average
-----	-----	-----	-----	-----
ANALYST	3000	3000	6000	3000
CLERK	1300	800	4150	1038
MANAGER	2975	2450	8275	2758
PRESIDENT	5000	5000	5000	5000
SALESMAN	1600	1250	5600	1400

6.Crie uma consulta para exibir o número de pessoas com o mesmo cargo.

JOB	COUNT (*)
-----	-----
ANALYST	2
CLERK	4
MANAGER	3
PRESIDENT	1
SALESMAN	4

Exercício 13 (continuação)

7. Determine o número de gerentes sem listá-los. Coloque um label na coluna Number of Managers.

```
Number of Managers
```

```
-----
```

6

8. Crie uma consulta para exibir a diferença entre os maiores e menores salários. Coloque um label na coluna DIFFERENCE.

```
DIFFERENCE
```

```
-----
```

4200

Se você tiver tempo, complete os exercícios abaixo:

9. Exiba o número do gerente e o salário do funcionário com menor pagamento sob a supervisão desse gerente. Exclua todos cujo gerente não seja conhecido. Exclua todos os grupos em que o salário mínimo seja menor do que US\$ 1.000. Classifique a saída em ordem decrescente de salário.

MGR	MIN (SAL)
-----	-----
7566	3000
7839	2450
7782	1300
7788	1100

10. Crie uma consulta para exibir o nome do departamento, o nome do local, o número de funcionários e o salário médio de todos os funcionários nesse departamento. Coloque um label nas colunas dname, loc, Number of People e Salary, respectivamente. Arredonde o salário médio para duas casas decimais

DNAME	LOC	Number of People	Salary
-----	-----	-----	-----
ACCOUNTING	NEW YORK	3	2916.67
RESEARCH	DALLAS	5	2175
SALES	CHICAGO	6	1566.67

Exercício 13 (continuação)

11. Crie uma consulta que exiba o número total de funcionários e, desse total, o número total de funcionários contratados em 1980, 1981, 1982 e 1983. Coloque os cabeçalhos apropriados nas colunas.

TOTAL	1980	1981	1982	1983
-----	-----	-----	-----	-----
14	1	10	2	1

12. Crie uma consulta matriz para exibir o cargo, o salário desse cargo baseado no número do departamento e o salário total desse cargo para todos os departamentos, colocando em cada coluna um cabeçalho apropriado.

Job	Dept 10	Dept 20	Dept 30	Total
-----	-----	-----	-----	-----
ANALYST		6000		6000
CLERK	1300	1900	950	4150
MANAGER	2450	2975	2850	8275
PRESIDENT	5000			5000
SALESMAN			5600	5600

