



Operadores de Conjuntos

Objetivos

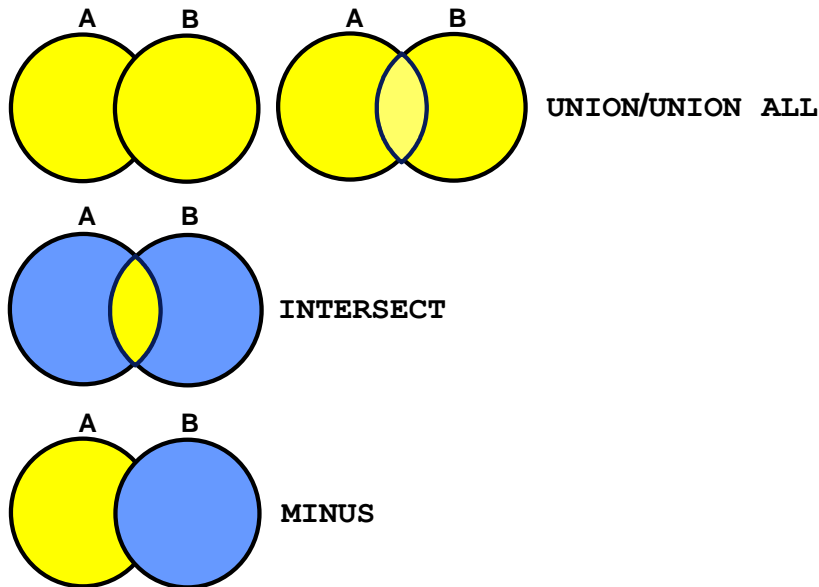
Ao concluir esta lição, você será capaz de:

- Descrever os operadores de conjunto
- Usar um operador de conjunto para combinar várias consultas em uma só
- Controlar a ordem das linhas retornadas

Objetivos

Nesta lição, você aprenderá a criar consultas com os operadores de conjunto.

Operadores de Conjunto



1-4

Operadores de Conjunto

Os operadores de conjunto combinam os resultados de duas ou mais consultas componentes em um resultado. As consultas que contêm operadores de conjunto são denominadas *consultas compostas*.

Operador	Retorna
UNION	Todas as linhas distintas selecionadas por qualquer uma das consultas
UNION ALL	Todas as linhas selecionadas por qualquer uma das consultas, inclusive as linhas duplicadas
INTERSECT	Todas as linhas distintas selecionadas pelas duas consultas
MINUS	Todas as linhas distintas selecionadas pela primeira instrução SELECT e não selecionadas na segunda instrução SELECT

Todos os operadores de conjunto têm a mesma precedência. Se uma instrução SQL contiver vários operadores de conjunto, o servidor Oracle os avaliará da esquerda (superior) para a direita (inferior) caso não haja parênteses especificando explicitamente outra ordem. Use parênteses para especificar explicitamente a ordem de avaliação nas consultas que utilizam o operador `INTERSECT` com outros operadores de conjunto.

Tabelas Usadas Nesta Lição

As tabelas usadas nesta lição são:

- **EMPLOYEES:** Fornece detalhes sobre todos os funcionários atuais
- **JOB_HISTORY:** Registra os detalhes relativos às datas inicial e final do cargo antigo, ao número de identificação do cargo e ao departamento quando um funcionário muda de cargo

Tabelas Usadas Nesta Lição

Duas tabelas são usadas nesta lição: `EMPLOYEES` e `JOB_HISTORY`.

A tabela `EMPLOYEES` armazena detalhes dos funcionários. Nos registros de recursos humanos, essa tabela armazena um número de identificação exclusivo e um endereço de e-mail para cada funcionário. Também são armazenados os detalhes do número de identificação de cargo, salário e gerente do funcionário. Além do salário, alguns funcionários recebem comissão, e essa informação também é rastreada. A empresa organiza as atribuições dos funcionários em cargos. Alguns funcionários trabalham há muito tempo na empresa e passaram por cargos diferentes. Essas informações são monitoradas com a tabela `JOB_HISTORY`. Quando um funcionário muda de cargo, os detalhes relativos às datas inicial e final do cargo antigo, ao número de identificação do cargo e ao departamento são registrados na tabela `JOB_HISTORY`.

A estrutura e os dados das tabelas `EMPLOYEES` e `JOB_HISTORY` são mostrados nas páginas seguintes.

Tabelas Usadas Nesta Lição (continuação)

Houve casos na empresa de pessoas que ocuparam o mesmo cargo mais de uma vez durante a sua permanência nesse estabelecimento. Por exemplo, considere o funcionário Taylor, admitido na empresa em 24-MAR-1998. Taylor ocupou o cargo SA_REP de 24-MAR-98 a 31-DEC-98 e o cargo SA_MAN de 01-JAN-99 a 31-DEC-99. Ele voltou para o cargo SA_REP, que é o seu cargo atual.

Da mesma forma, considere o funcionário Whalen, admitido na empresa em 17-SEP-1987. Whalen ocupou o cargo AD_ASST de 17-SEP-87 a 17-JUN-93 e o cargo AC_ACCOUNT de 01-JUL-94 a 31-DEC-98. Ele voltou para o cargo AD_ASST, que é o seu cargo atual.

DESCRIBE employees

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)
DEPARTMENT_NAME		VARCHAR2(14)

Tabelas Usadas Nesta Lição (continuação)

```
SELECT employee_id, last_name, job_id, hire_date,
       department_id
FROM employees;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE	DEPARTMENT_ID
100	King	AD_PRES	17-JUN-87	90
101	Kochhar	AD_VP	21-SEP-89	90
102	De Haan	AD_VP	13-JAN-93	90
103	Hunold	IT_PROG	03-JAN-90	60
104	Ernst	IT_PROG	21-MAY-91	60
107	Lorentz	IT_PROG	07-FEB-99	60
124	Mourgos	ST_MAN	16-NOV-99	50
141	Rajs	ST_CLERK	17-OCT-95	50
142	Davies	ST_CLERK	29-JAN-97	50
143	Matos	ST_CLERK	15-MAR-98	50
144	Vargas	ST_CLERK	09-JUL-98	50
149	Zlotkey	SA_MAN	29-JAN-00	80
174	Abel	SA_REP	11-MAY-96	80
176	Taylor	SA_REP	24-MAR-98	80
EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE	DEPARTMENT_ID
178	Grant	SA_REP	24-MAY-99	
200	Whalen	AD_ASST	17-SEP-87	10
201	Hartstein	MK_MAN	17-FEB-96	20

...

20 rows selected.

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
START_DATE	NOT NULL	DATE
END_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
DEPARTMENT_ID		NUMBER(4)

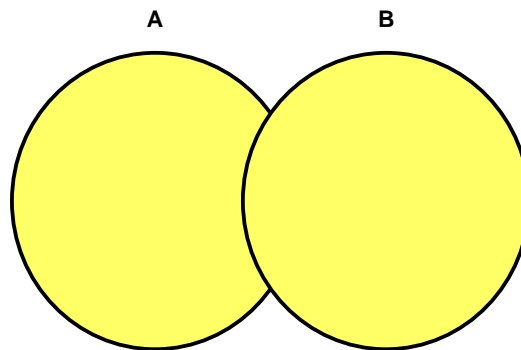
Tabelas Usadas Nesta Lição (continuação)

```
SELECT * FROM job_history;
```

EMPLOYEE_ID	START_DAT	END_DATE	JOB_ID	DEPARTMENT_ID
102	13-JAN-93	24-JUL-98	IT_PROG	60
101	21-SEP-89	27-OCT-93	AC_ACCOUNT	110
101	28-OCT-93	15-MAR-97	AC_MGR	110
201	17-FEB-96	19-DEC-99	MK_REP	20
114	24-MAR-98	31-DEC-99	ST_CLERK	50
122	01-JAN-99	31-DEC-99	ST_CLERK	50
200	17-SEP-87	17-JUN-93	AD_ASST	90
176	24-MAR-98	31-DEC-98	SA_REP	80
176	01-JAN-99	31-DEC-99	SA_MAN	80
200	01-JUL-94	31-DEC-98	AC_ACCOUNT	90

10 rows selected.

Operador UNION



O operador `UNION` retorna resultados das duas consultas após eliminar os valores duplicados.

1-9

Operador `UNION`

O operador `UNION` retorna todas as linhas selecionadas pelas consultas. Utilize-o para retornar todas as linhas de várias tabelas e eliminar linhas duplicadas.

Diretrizes

O número de colunas e os tipos de dados das colunas selecionadas devem ser idênticos em todas as instruções `SELECT` usadas na consulta. Os nomes das colunas não precisam ser idênticos.

`UNION` opera em todas as colunas selecionadas.

Os valores nulos não são ignorados durante a verificação de valores duplicados.

O operador `IN` tem precedência em relação ao operador `UNION`.

Por default, a saída é classificada em ordem crescente da primeira coluna da cláusula `SELECT`.

Usando o Operador UNION

Exiba os detalhes dos cargos atual e anterior de todos os funcionários. Exiba cada funcionário apenas uma vez.

```
SELECT employee_id, job_id
FROM employees
UNION
SELECT employee_id, job_id
FROM job_history;
```

EMPLOYEE_ID	JOB_ID
1	100AD PRES
2	101AC ACCOUNT
...	
87	180SH CLERK
88	181SH CLERK
89	182SH CLERK
90	183SH CLERK
...	
114	205AC MGR
115	206AC ACCOUNT

1-10

Usando o Operador UNION

O operador UNION elimina registros duplicados. Se houver registros idênticos nas tabelas EMPLOYEES e JOB_HISTORY, eles serão exibidos apenas uma vez. Na saída mostrada no slide, observe que o registro para o funcionário com EMPLOYEE_ID 200 aparece duas vezes, já que o valor em JOB_ID é diferente em cada linha.

Considere o seguinte exemplo:

```
SELECT employee_id, job_id, department_id
FROM employees
UNION
SELECT employee_id, job_id, department_id
FROM job_history;
```

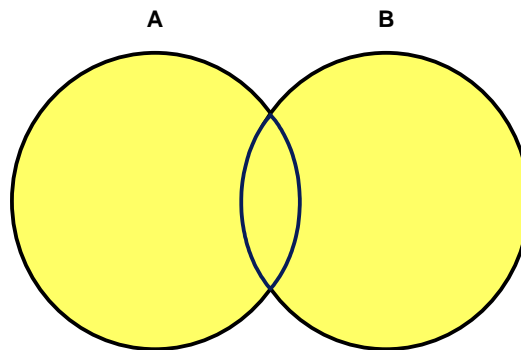
EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
...		
200	AC_ACCOUNT	90
200	AD_ASST	10
200	AD_ASST	90
...		

29 rows selected.

Usando o Operador UNION (continuação)

Na saída anterior, o funcionário 200 aparece três vezes. Por quê? Observe os valores de `DEPARTMENT_ID` relativos a esse funcionário. O valor de `DEPARTMENT_ID` em uma linha é 90, em outra é 10 e na terceira é 90. Em função dessas combinações exclusivas de IDs de cargo e IDs de departamento, cada linha relativa ao funcionário 200 é exclusiva e, por isso, não é considerada duplicada. Observe que a saída é classificada em ordem crescente da primeira coluna da cláusula `SELECT` (nesse caso, `EMPLOYEE_ID`).

Operador UNION ALL



O operador **UNION ALL** retorna resultados das duas consultas, inclusive todos os valores duplicados.

1-12

Operador UNION ALL

Use o operador **UNION ALL** para retornar todas as linhas de várias consultas.

Diretrizes

As diretrizes para **UNION** e **UNION ALL** são as mesmas, apenas com duas exceções relativas a **UNION ALL**:

Ao contrário de **UNION**, as linhas duplicadas não são eliminadas e, por default, a saída não é classificada.

A palavra-chave **DISTINCT** não pode ser usada.

Usando o Operador UNION ALL

Exiba os departamentos anterior e atual de todos os funcionários.

```
SELECT employee_id, job_id, department_id
FROM employees
UNION ALL
SELECT employee_id, job_id, department_id
FROM job_history
ORDER BY employee_id;
```

	EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
1	100	AD PRES	90
2	101	AD VP	90
...			
108	200	AD ASST	10
109	200	AD ASST	90
110	200	AC ACCOUNT	90
...			
116	205	AC MGR	110
117	206	AC ACCOUNT	110

1-13

Operador UNION ALL (continuação)

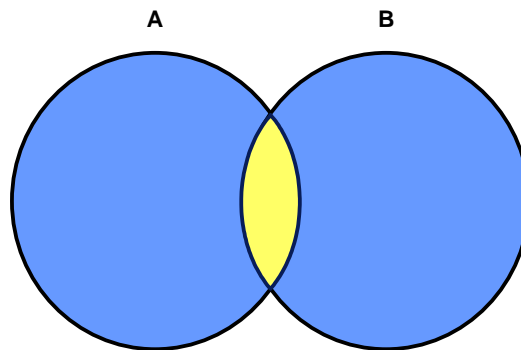
No exemplo, são selecionadas 30 linhas. A combinação das duas tabelas totaliza 30 linhas. O operador UNION ALL não elimina linhas duplicadas. UNION retorna todas as linhas distintas selecionadas pelas consultas. UNION ALL retorna todas as linhas selecionadas pelas consultas, inclusive todas as duplicadas. Considere a consulta do slide, criada agora com a cláusula UNION:

```
SELECT employee_id, job_id, department_id
FROM employees
UNION
SELECT employee_id, job_id, department_id
FROM job_history
ORDER BY employee_id;
```

A consulta anterior retorna 29 linhas. Isso porque ela elimina a seguinte linha (já que ela é duplicada):

EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
176	SA_REP	80

Operador INTERSECT



O operador `INTERSECT` retorna linhas comuns às duas consultas.

1-14

Operador `INTERSECT`

Use o operador `INTERSECT` para retornar todas as linhas comuns a várias consultas.

Diretrizes

O número de colunas e os tipos de dados das colunas selecionadas pelas instruções `SELECT` nas consultas devem ser idênticos em todas as instruções `SELECT` usadas na consulta. Os nomes das colunas não precisam ser idênticos.

O resultado não é alterado quando é invertida a ordem das tabelas de interseção.

`INTERSECT` não ignora valores `nulos`.

Usando o Operador INTERSECT

Exiba os IDs de funcionário e os IDs de cargo dos funcionários que, no momento, estão no mesmo cargo que ocupavam quando foram admitidos pela empresa (ou seja, eles mudaram de cargo, mas agora voltaram para o cargo original).

```
SELECT employee_id, job_id
FROM employees
INTERSECT
SELECT employee_id, job_id
FROM job_history;
```

	EMPLOYEE_ID	JOB_ID
1	176	SA_REP
2	200	AD_ASST

1-15

Operador INTERSECT (continuação)

No exemplo do slide, a consulta retorna apenas os registros com os mesmos valores nas colunas selecionadas das duas tabelas.

Quais serão os resultados se você adicionar a coluna `DEPARTMENT_ID` à instrução `SELECT` da tabela `EMPLOYEES`, adicionar a coluna `DEPARTMENT_ID` à instrução `SELECT` da tabela `JOB_HISTORY` e executar essa consulta? Os resultados poderão ser diferentes em função da inclusão de outra coluna cujos valores podem ou não ser duplicados.

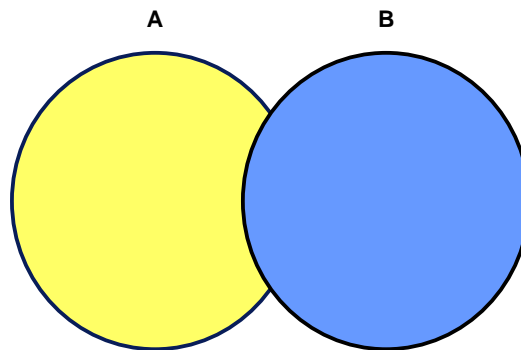
Exemplo

```
SELECT employee_id, job_id, department_id
FROM employees
INTERSECT
SELECT employee_id, job_id, department_id
FROM job_history;
```

EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
176	SA_REP	80

O funcionário 200 não está mais incluído nos resultados, pois o valor de `EMPLOYEES.DEPARTMENT_ID` é diferente do valor de `JOB_HISTORY.DEPARTMENT_ID`.

Operador MINUS



O operador **MINUS** retorna as linhas da primeira consulta que não estão presentes na segunda consulta.

1-16

Operador MINUS

Use o operador **MINUS** para exibir as linhas retornadas pela primeira consulta que não estão presentes na segunda consulta (a primeira instrução **SELECT** subtraída da segunda instrução **SELECT**).

Diretrizes

O número de colunas e os tipos de dados das colunas selecionadas pelas instruções **SELECT** nas consultas devem ser idênticos em todas as instruções **SELECT** usadas na consulta. Os nomes das colunas não precisam ser idênticos.

Todas as colunas da cláusula **WHERE** devem estar presentes na cláusula **SELECT** para que o operador **MINUS** seja executado.

Operador MINUS

Exiba os IDs dos funcionários que nunca mudaram de cargo.

```
SELECT employee_id, job_id
FROM employees
MINUS
SELECT employee_id, job_id
FROM job_history;
```

	EMPLOYEE_ID	JOB_ID
1	100	AD PRES
2	101	AD VP
3	102	AD VP
4	103	IT PROG
...		
104	205	AC MGR
105	206	AC ACCOUNT

Operador MINUS (continuação)

No exemplo do slide, os IDs de funcionário e os IDs de cargo na tabela `JOB_HISTORY` são subtraídos dos IDs correspondentes na tabela `EMPLOYEES`. O conjunto de resultados exibe os funcionários restantes após a subtração; eles são representados pelas linhas existentes na tabela `EMPLOYEES`, mas que não estão presentes na tabela `JOB_HISTORY`. Esses registros são relativos aos funcionários que nunca mudaram de cargo.

Diretrizes de Operadores de Conjunto

- As expressões das listas **SELECT** devem corresponder em número e tipo de dados.
- É possível usar parênteses para alterar a seqüência de execução.
- A cláusula **ORDER BY**:
 - Só pode aparecer no final da instrução
 - Aceitará o nome da coluna, os apelidos da primeira instrução **SELECT** ou a notação posicional

1-18

Diretrizes de Operadores de Conjunto

As expressões nas listas de seleção das consultas devem corresponder em número e tipo de dados. As consultas que usam operadores **UNION**, **UNION ALL**, **INTERSECT** e **MINUS** na cláusula **WHERE** devem ter o mesmo número e tipo das colunas da lista **SELECT**. Por exemplo:

```
SELECT employee_id, department_id
FROM employees
WHERE (employee_id, department_id)
      IN (SELECT employee_id, department_id
          FROM employees
          UNION
          SELECT employee_id, department_id
          FROM job_history);
```

A cláusula **ORDER BY**:

Só pode aparecer no final da instrução

Aceitará o nome da coluna, um apelido ou a notação posicional

O nome ou o apelido da coluna, se usado em uma cláusula **ORDER BY**, deverá originar-se da primeira lista **SELECT**.

É possível usar operadores de conjunto em subconsultas.

O Servidor Oracle e os Operadores de Conjunto

- As linhas duplicadas são eliminadas automaticamente, exceto em `UNION ALL`.
- Os nomes das colunas da primeira consulta aparecem no resultado.
- Por default, a saída é classificada em ordem crescente, exceto em `UNION ALL`.

O Servidor Oracle e os Operadores de Conjunto

Quando uma consulta usa operadores de conjunto, o servidor Oracle elimina as linhas duplicadas automaticamente, exceto no caso do operador `UNION ALL`. Os nomes das colunas na saída são decididos pela lista de colunas da primeira instrução `SELECT`. Por default, a saída é classificada em ordem crescente da primeira coluna da cláusula `SELECT`.

As expressões correspondentes nas listas de seleção das consultas componentes de uma consulta composta devem coincidir em número e tipo de dados. Se as consultas componentes selecionarem dados de caractere, o tipo de dados dos valores retornados será determinado da seguinte maneira:

Se as duas consultas selecionarem valores do tipo de dados `CHAR`, os valores retornados terão esse tipo de dados.

Se uma ou as duas consultas selecionarem valores do tipo de dados `VARCHAR2`, os valores retornados terão esse tipo de dados.

Correspondência entre Instruções SELECT

Com o operador UNION, exiba o ID de departamento, a localização e a data de admissão de todos os funcionários.

```
SELECT department_id, TO_NUMBER(null)
       location, hire_date
FROM   employees
UNION
SELECT department_id, location_id, TO_DATE(null)
FROM   departments;
```

	DEPARTMENT_ID	LOCATION	HIRE_DATE
1	10	1700	(null)
2	10	(null)	17/09/87
3	20	1800	(null)
4	20	(null)	17/02/96
5	20	(null)	17/08/97
6	30	1700	(null)
...			
128	260	1700	(null)
129	270	1700	(null)
130	(null)	(null)	24/05/99

1-20

Correspondência entre Instruções SELECT

Como as expressões nas listas de seleção das consultas devem corresponder em número, você pode usar colunas fictícias e as functions de conversão de tipos de dados para seguir essa regra. No slide, a localização do nome é fornecida como o cabeçalho da coluna fictícia. A function `TO_NUMBER` é usada na primeira consulta para corresponder ao tipo de dados `NUMBER` da coluna `LOCATION_ID` recuperada pela segunda consulta. Da mesma forma, a function `TO_DATE` na segunda consulta é usada para corresponder ao tipo de dados `DATE` da coluna `HIRE_DATE` recuperada pela primeira consulta.

Correspondência entre Instruções

SELECT:

Exemplo

Com o operador **UNION**, exiba o ID de funcionário, o ID do cargo e o salário de todos os funcionários.

```
SELECT employee_id, job_id, salary
FROM employees
UNION
SELECT employee_id, job_id, 0
FROM job_history;
```

	EMPLOYEE_ID	JOB_ID	SALARY
1	100	AD PRES	24000
2	101	AC ACCOUNT	0
3	101	AC MGR	0
4	101	AD VP	17000
...			
115	204	PR REP	10000
116	205	AC MGR	12000
117	206	AC ACCOUNT	8300

1-21

Correspondência entre Instruções SELECT: Exemplo

As tabelas **EMPLOYEES** e **JOB_HISTORY** têm várias colunas em comum (por exemplo, **EMPLOYEE_ID**, **JOB_ID** e **DEPARTMENT_ID**). Mas e se você quiser que a consulta exiba o ID de funcionário, o ID de cargo e o salário com o operador **UNION**, sabendo que o salário existe apenas na tabela **EMPLOYEES**?

O exemplo de código do slide estabelece a correspondência entre as colunas **EMPLOYEE_ID** e **JOB_ID** das tabelas **EMPLOYEES** e **JOB_HISTORY**. Um valor literal **0** é adicionado à instrução **SELECT** de **JOB_HISTORY** para corresponder à coluna numérica **SALARY** na instrução **SELECT** de **EMPLOYEES**.

Nos resultados anteriores, cada linha da saída que corresponde a um registro da tabela **JOB_HISTORY** contém **0** na coluna **SALARY**.

O Comando *SQL* COLUMN (continuação)

Sintaxe:

```
COL[UMN] [{column|alias} [option]]
```

Em que OPTION é:

CLE[AR]: Remove todos os formatos de coluna

HEA[DING] *texto*: Define o cabeçalho da coluna

FOR[MAT] *formato*: Altera a exibição da coluna usando um modelo de formato

NOPRINT | PRINT: Suprime ou exhibe os dados e os cabeçalhos de colunas

NULL

A instrução a seguir suprime os dados e o cabeçalho da coluna A_DUMMY. Observe que a primeira cláusula SELECT do slide anterior cria uma coluna fictícia denominada A_DUMMY.

```
COLUMN a_dummy NOPRINT
```

Sumário

Nesta lição, você aprendeu a:

- Usar **UNION** para retornar todas as linhas distintas
- Usar **UNION ALL** para retornar todas as linhas, inclusive as duplicadas
- Usar **INTERSECT** para retornar todas as linhas compartilhadas pelas duas consultas
- Usar **MINUS** para retornar todas as linhas distintas selecionadas pela primeira consulta, mas não pela segunda
- Usar **ORDER BY** somente no final da instrução

1-23

Sumário

O operador **UNION** retorna todas as linhas selecionadas pelas consultas. Utilize-o para retornar todas as linhas de várias tabelas e eliminar linhas duplicadas.

Use o operador **UNION ALL** para retornar todas as linhas de várias consultas. Ao contrário do que ocorre com o operador **UNION**, as linhas duplicadas não são eliminadas e, por default, a saída não é classificada.

Use o operador **INTERSECT** para retornar todas as linhas comuns a várias consultas.

Use o operador **MINUS** para exibir as linhas retornadas pela primeira consulta que não estão presentes na segunda consulta.

Lembre-se de usar a cláusula **ORDER BY** somente no final da instrução composta.

Certifique-se de que as expressões correspondentes nas listas **SELECT** coincidem em número e tipo de dados.

Exercício: Visão Geral

Neste exercício, você usará os operadores de conjunto para criar relatórios:

- Usando o operador `UNION`
- Usando o operador `INTERSECTION`
- Usando o operador `MINUS`

Exercício: Visão Geral

Neste exercício, você criará consultas com os operadores de conjunto.

Exercício

1. O departamento de recursos humanos precisa de uma lista de IDs dos departamentos que não contêm o ID de cargo ST_CLERK. Use os operadores de conjunto para criar esse relatório.

DEPARTMENT_ID
10
20
60
80
90
110
190

7 rows selected.

2. O departamento de recursos humanos precisa de uma lista de países nos quais não há departamentos. Exiba o ID e o nome dos países. Use os operadores de conjunto para criar esse relatório.

CO	COUNTRY_NAME
DE	Germany

3. Produza uma lista de cargos dos departamentos 10, 50 e 20, nessa ordem. Exiba o ID de cargo e o ID de departamento usando operadores de conjunto.

JOB_ID	DEPARTMENT_ID
AD_ASST	10
ST_CLERK	50
ST_MAN	50
MK_MAN	20
MK_REP	20

4. Crie um relatório que liste os IDs de funcionário e os IDs de cargo dos funcionários que, no momento, estão no mesmo cargo que ocupavam quando foram admitidos pela empresa (ou seja, eles mudaram de cargo, mas agora voltaram para o cargo original).

EMPLOYEE_ID	JOB_ID
176	SA_REP
200	AD_ASST

Exercício

5. O departamento de recursos humanos precisa de um relatório com as seguintes especificações:

Sobrenome e ID do departamento de todos os funcionários da tabela

EMPLOYEES, mesmo que não pertençam a um departamento

ID e nome de todos os departamentos da tabela DEPARTMENTS, mesmo que não tenham funcionários

Crie uma consulta composta para isso.

LAST_NAME	DEPARTMENT_ID	TO_CHAR(NULL)
Abel	80	
Davies	50	
De Haan	90	
Ernst	60	
Fay	20	
Gietz	110	
Grant		
Hartstein	20	
Higgins	110	
Hunold	60	
King	90	
Kochhar	90	
Lorentz	60	
Matos	50	
LAST_NAME	DEPARTMENT_ID	TO_CHAR(NULL)
Mourgos	50	
Rajs	50	
Taylor	80	
Vargas	50	
Whalen	10	
Zlotkey	80	
	10	Administration
	20	Marketing
	50	Shipping
	60	IT
	80	Sales
	90	Executive
	110	Accounting
	190	Contracting

Bibliografia Utilizada

Database SQL Language Reference: <http://docs.oracle.com/database/121/SQLRF/toc.htm>

Manuais Oracle – Oracle Database 12c: SQL Workshop I/II

*Esta apresentação possui material de referência com propriedade da Oracle.
Copyright © , Oracle. Todos os direitos reservados.*



OBRIGADO



profalexandre.barcelos@fiap.com.br



<https://www.linkedin.com/in/alexandrebarcelos>

FIAP

Copyright © 2023 | Professor Me. Alexandre Barcelos
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente
proibido sem consentimento formal, por escrito, do professor/autor.

