

# ORACLE

## Academy

# Database Programming with SQL

7-1

## Equijunção Oracle e Produto Cartesiano

**ORACLE**  
Academy



# Objetivos

- Nesta lição, você aprenderá a:
  - Nomear as junções proprietárias da Oracle e seus equivalentes no ANSI/ISO SQL: 99
  - Construir e executar uma instrução SELECT que resulte em um produto cartesiano
  - Construir e executar instruções SELECT para acessar dados de mais de uma tabela usando uma equijunção
  - Construir e executar instruções SELECT que adicionem condições de pesquisa usando o operador AND
  - Aplicar a regra para usar aliases de tabela em uma instrução de junção

# Finalidade

- A seção anterior abordou a consulta e o retorno de dados de mais de uma tabela em um banco de dados relacional usando a sintaxe do ANSI/ISO SQL: 99
- As versões legadas dos bancos de dados Oracle exigiam junções para usar a sintaxe de junção Proprietária da Oracle, e muitos desses bancos de dados mais antigos ainda são usados
- Esta lição apresenta a sintaxe de junção Proprietária da Oracle para Equijunções e Produto Cartesiano e seus equivalentes no ANSI/ISO SQL: 99

# Comandos de Junção

- Os dois conjuntos de comandos ou sintaxes que podem ser usados para fazer ligações entre as tabelas em um banco de dados:
  - Junções proprietárias da Oracle
  - Junções em conformidade com o padrão ANSI/ISO SQL: 99



# Comparação entre Junções

- Comparando Junções Proprietárias da Oracle com Junções do ANSI/ISO SQL: 1999

Junção Proprietária da Oracle	Equivalente no ANSI/ISO SQL: 1999
Produto Cartesiano	CROSS JOIN
Equijunção	NATURAL JOIN  Cláusula JOIN USING  Cláusula JOIN ON (se o operador de igualdade for usado)
Não-equijunção	Cláusula ON

# Junções Proprietárias da ORACLE

- Para consultar dados de mais de uma tabela usando a sintaxe proprietária da Oracle, use uma condição de junção na cláusula WHERE
- O formato básico de uma instrução de junção é:

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```

# Junções Proprietárias da ORACLE

- Imagine dois alunos com o mesmo sobrenome na mesma sala de aula
- Quando precisa falar com o "Jackson", a professora esclarece a qual dos dois se refere mencionando o nome antes do sobrenome
- Para facilitar a leitura de uma instrução de junção e acelerar o acesso ao banco de dados, recomenda-se mencionar o nome da tabela antes do nome da coluna

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```

# Junções Proprietárias da ORACLE

- Isso é chamado de "qualificação de colunas"
- A combinação do nome da tabela e do nome da coluna ajuda a eliminar ambiguidades quando duas tabelas contêm uma coluna com o mesmo nome
- Quando o mesmo nome de coluna aparece em ambas as tabelas, o nome da tabela precisa vir antes do nome da coluna

# Exemplo de Sintaxe de Junção

- Para qualificar as colunas, usa-se a sintaxe `tablename.columnname`, como mostrado no exemplo abaixo

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```

# Equijunção

- Às vezes chamada de junção "simples" ou "interna", uma equijunção é uma junção de tabelas que combina linhas que tenham os mesmos valores para as colunas especificadas
- No ANSI, equijunção é equivalente a:
  - NATURAL JOIN
  - JOIN USING
  - JOIN ON (quando a condição de junção usar "=")
- O slide a seguir demonstra o quê, onde e como da junção das tabelas

# Equijunção

- O quê? A cláusula SELECT especifica os nomes de coluna a serem exibidos
- Onde? A cláusula FROM especifica as tabelas que o banco de dados deve acessar, separadas por vírgulas.
- Como? A cláusula WHERE especifica como as tabelas devem ser unidas.
- Uma Equijunção usa o operador de igual a para especificar a condição de junção

# Equijunção

SELECT employees.last\_name, employees.job\_id, jobs.job\_title O quê?

FROM employees, jobs Onde?

WHERE employees.job\_id = jobs.job\_id; Como?

LAST_NAME	JOB_ID	JOB_TITLE
King	AD_PRES	President
Kochhar	AD_VP	Administration Vice President
De Haan	AD_VP	Administration Vice President
Whalen	AD_ASST	Administration Assistant
Higgins	AC_MGR	Accounting Manager
Gietz	AC_ACCOUNT	Public Accountant
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
...	...	...

# Equijunção

- Outro exemplo:

```
SELECT employees.last_name, departments.department_name  
FROM employees, departments  
WHERE employees.department_id = departments.department_id;
```

LAST_NAME	DEPARTMENT_NAME
Whalen	Administration
Hartstein	Marketing
Fay	Marketing
Mourgos	Shipping
Rajs	Shipping
Davies	Shipping
Matos	Shipping
...	...

# Aliases

- Pode ser complicado trabalhar com nomes longos de colunas e tabelas
- Felizmente, existe uma maneira de encurtar a sintaxe: usar aliases
- Para distinguir colunas que têm nomes idênticos, mas residem em tabelas diferentes, use aliases de tabela
- Um alias de tabela é semelhante a um alias de coluna, ou seja, renomeia um objeto em uma instrução
- Ele é criado com a inclusão do novo nome para a tabela logo após o nome da tabela na cláusula FROM

# Aliases de Tabela

- A consulta abaixo usa aliases de tabela

```
SELECT last_name, e.job_id, job_title  
FROM employees e, jobs j  
WHERE e.job_id = j.job_id  
AND department_id = 80;
```

LAST_NAME	JOB_ID	JOB_TITLE
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
Taylor	SA_REP	Sales Representative

- Quando os nomes das colunas não estão duplicados em duas tabelas, você não precisa adicionar o nome ou alias da tabela ao nome da coluna

# Aliases de Tabela

- Se for usado na cláusula FROM, o alias de tabela deverá ser substituído pelo nome da tabela em toda a instrução SELECT
- O uso do nome de uma tabela na cláusula SELECT que recebeu um alias na cláusula FROM resultará em um erro

```
SELECT last_name, employees.job_id, job_title  
FROM employees e, jobs j  
WHERE e.job_id = j.job_id  
AND department_id = 80;
```



ORA-00904: "EMPLOYEES"."JOB\_ID": invalid identifier

# Junção de Produtos Cartesianos

- Se duas tabelas em uma consulta de junção não tiverem uma condição de junção especificada na cláusula WHERE ou se a condição de junção for inválida, o Servidor Oracle retorna o produto cartesiano das duas tabelas
- Ele é uma combinação de cada linha de uma tabela com cada linha da outra tabela
- Um produto cartesiano é equivalente ao CROSS JOIN do ANSI
- Para evitar a obtenção de um produto cartesiano, inclua sempre uma condição de junção válida na cláusula WHERE

# Junção de Produtos Cartesianos

- Nesta consulta, a condição de junção foi omitida:

```
SELECT employees.last_name, departments.department_name  
FROM employees, departments;
```

LAST_NAME	DEPARTMENT_NAME
Abel	Administration
Davies	Administration
De Haan	Administration
Ernst	Administration
Fay	Administration
Gietz	Administration
Grant	Administration
...	...

160 rows returned in 0.01 seconds

# Restringindo Linhas em uma Junção

- Como nas consultas com uma única tabela, a cláusula WHERE pode ser usada para restringir as linhas consideradas em uma ou mais tabelas da junção
- A consulta mostrada usa o operador AND para restringir as linhas retornadas

```
SELECT employees.last_name, employees.job_id, jobs.job_title  
FROM employees, jobs  
WHERE employees.job_id = jobs.job_id  
AND employees.department_id = 80;
```

LAST_NAME	JOB_ID	JOB_TITLE
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Perwakilan Penjualan
Taylor	SA_REP	Perwakilan Penjualan

# Exemplo de Sintaxe de Junção

- Se você quisesse unir três tabelas, quantas junções seriam necessárias?
- Quantas pontes são necessárias para unir três ilhas?
- Para unir três tabelas, você precisa adicionar outra condição de junção à cláusula WHERE usando o operador AND

Tabela 1

Tabela 2

Tabela 3

# Exemplo de Sintaxe de Junção

- Vamos supor que precisamos de um relatório sobre os nossos funcionários e a cidade onde o departamento deles está localizado
- Precisamos juntar três tabelas: as de funcionários, departamentos e locais

```
SELECT last_name, city
FROM employees e, departments d,
     locations l
WHERE e.department_id = d.department_id
  AND d.location_id = l.location_id;
```

LAST_NAME	CITY
Hartstein	Toronto
Fay	Toronto
Zlotkey	Oxford
Abel	Oxford
...	...

# Terminologia

- Estes são os principais termos usados nesta lição:
  - Alias
  - Produto Cartesiano
  - Equijunção
  - Condições de Junção
  - Junção Proprietária

# Resumo

- Nesta lição, você aprendeu a:
  - Nomear as junções proprietárias da Oracle e seus equivalentes no ANSI/ISO SQL: 99
  - Construir e executar uma instrução SELECT que resulte em um produto cartesiano
  - Construir e executar instruções SELECT para acessar dados de mais de uma tabela usando uma equijunção
  - Construir e executar instruções SELECT que adicionem condições de pesquisa usando o operador AND
  - Aplicar a regra para usar aliases de tabela em uma instrução de junção

# **ORACLE**

## Academy