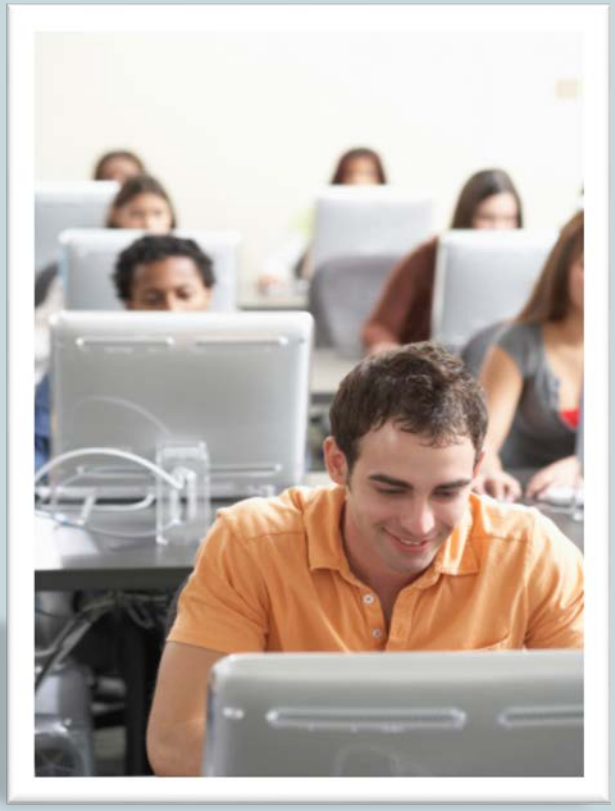

ORACLE
Academy

Programação de Banco de Dados com SQL

12-1
Instruções INSERT



ORACLE Academy

Objetivos

Nesta lição, você aprenderá a:

- Explicar a importância de ser capaz de alterar os dados em um banco de dados
- Construir e executar instruções INSERT que inserem uma única linha usando uma cláusula VALUES
- Construir e executar instruções INSERT que usam valores especiais, nulos e de data
- Construir e executar instruções INSERT que copiam linhas de uma tabela para outra usando uma subconsulta

Finalidade

- Até agora, você aprendeu como acessar dados em um banco de dados.
- Está na hora de aprender como fazer mudanças nesses dados.
- Nos negócios, os bancos de dados são dinâmicos.
- Eles passam por processos constantes de inserção, atualização e exclusão de dados.

Finalidade

- Pense em quantas vezes o banco de dados de alunos de uma escola muda diária e anualmente.
- A não ser que sejam feitas mudanças, o banco de dados logo perderá sua utilidade.
- Nesta lição, você começará a usar instruções com data manipulation language (DML) para fazer alterações em um banco de dados.

Copie as Tabelas Antes da Inserção

- Você será responsável pela alteração de tabelas em seu esquema.
- Você também será responsável pela restauração delas, exatamente como se fosse o Administrador do Banco de Dados.
- Para manter as tabelas de seu esquema no estado original, você fará uma cópia de cada tabela antes de completar as atividades práticas nesta e nas próximas lições.
- Em cada atividade prática, você usará a cópia da tabela criada, e não a original.
- Se inadvertidamente alterar a cópia de uma tabela, você poderá usar a tabela original para restaurar a cópia.

Copie as Tabelas Antes da Inserção

- Você deve dar um nome a cada cópia da tabela: copy_nomedatabela.
- As cópias da tabela não herdarão as regras de integridade entre chave primária e estrangeira (constraints de relacionamento) associadas das tabelas originais.
- Já os tipos de dados das colunas serão herdados pelas cópias da tabela.

Sintaxe para Criar a Cópia de uma Tabela

- Sintaxe de criação de tabela:

```
CREATE TABLE copy_nomedatabela
 AS (SELECT * FROM nomedatabela);
```

- Por exemplo:

```
CREATE TABLE copy_employees
AS (SELECT * FROM employees);
```

```
CREATE TABLE copy_departments
AS (SELECT * FROM departments);
```

Sintaxe para Criar a Cópia de uma Tabela

- Para verificar e exibir a cópia da tabela, use as seguintes instruções DESCRIBE e SELECT:

```
DESCRIBE copy_employees;
```

```
SELECT * FROM copy_employees;
```

```
DESCRIBE copy_departments;
```

```
SELECT * FROM copy_departments;
```

INSERT

- A instrução INSERT é usada para adicionar uma nova linha a uma tabela. Essa instrução requer três valores:
  - o nome da tabela
  - os nomes das colunas na tabela a serem preenchidas
  - os valores correspondentes de cada coluna
- Como podemos inserir os dados abaixo para criar um novo departamento na tabela copy_departments?

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
200	Human Resources	205	1500

INSERT

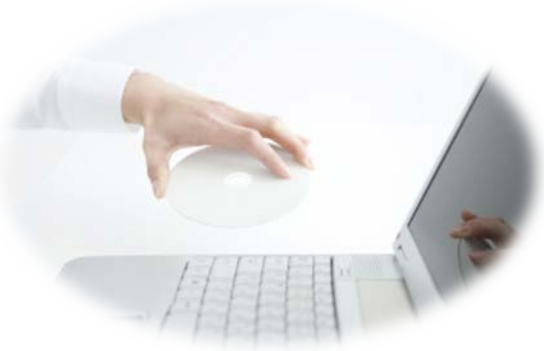
- A sintaxe abaixo usa INSERT para adicionar um novo departamento na tabela copy_departments.
- A instrução lista explicitamente as colunas conforme aparecem na tabela.
- Os valores de cada coluna são listados na mesma ordem.
  - Observe que os valores numéricos não ficam entre aspas

```
INSERT INTO copy_departments
 (department_id, department_name, manager_id, location_id)
VALUES
 (200,'Human Resources', 205, 1500);
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
200	Human Resources	205	1500

INSERT

- Outra maneira de inserir valores em uma tabela é adicioná-los implicitamente por meio da omissão dos nomes das colunas.
- Tenha cuidado: os valores de cada coluna devem ter exatamente a mesma ordem padrão na qual aparecem na tabela (como mostrado na instrução DESCRIBE), e um valor deve ser fornecido para cada coluna.



INSERT

- A instrução INSERT neste exemplo foi gravada sem nomear as colunas explicitamente.
- No entanto, para manter a clareza, é melhor usar os nomes das colunas em uma cláusula INSERT.

```
INSERT INTO copy_departments
VALUES
  (210,'Estate Management', 102, 1700);
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
210	Estate Management	102	1700

Verifique a Tabela Primeiro

- Antes de inserir os dados, você deve verificar vários detalhes da tabela.
- A instrução DESCRIBE nomedatabela retornará uma descrição da estrutura da tabela no gráfico de resumo.
- RESUMO DA TABELA COPY_DEPARTMENTS:

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
DEPARTMENT_NAME	VARCHAR2	30	-	-	-	-
MANAGER_ID	NUMBER	-	6	0	-	✓
LOCATION_ID	NUMBER	-	4	0	-	✓

Resumo da Tabela

- Como mostrado no exemplo, o resumo da tabela fornece informações sobre cada coluna, como:
  - a permissão de valores duplicados
  - o tipo de dados permitido
  - a quantidade de dados permitida
  - a permissão de valores nulos

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

Resumo da Tabela

- Observe que a coluna Data Type para os tipos de dados de caracteres especifica entre parênteses o número máximo de caracteres permitido.

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

Resumo da Tabela

- First_name tem o tipo de dados VARCHAR2(20). Isso significa que até 20 caracteres podem ser incluídos nessa coluna.

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

Resumo da Tabela

- Para os tipos de dados de número, os parênteses especificam Precision e Scale.
- Precision é o número total de dígitos e Scale é o número de dígitos à direita da casa decimal.

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

Resumo da Tabela

- A coluna SALARY permite números com Precision 8 e Scale 2.
- O valor máximo permitido nessa coluna é 999999.99.

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

Inserindo Linhas com Valores Nulos

- A instrução INSERT não precisa especificar todas as colunas: as anuláveis podem ser excluídas.
- Se for atribuído um valor a todas as colunas que o exigem, a inserção funcionará.

Inserindo Linhas com Valores Nulos

- No nosso exemplo, a coluna EMAIL é definida como NOT NULL.
- Uma tentativa implícita de adicionar valores à tabela, como mostrado, geraria um erro.

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, phone_number, hire_date,
 job_id, salary)
VALUES
  (302,'Grigorz','Polanski', '8586667641', '15-Jun-2015',
 'IT_PROG',4200);
```

```
ORA-01400: cannot insert NULL into
("US_A009EMEA815_PLSQL_T01"."COPY_EMPLOYEES"."EMAIL")
```

Inserindo Linhas com Valores Nulos

- Uma inserção implícita colocará automaticamente um valor nulo em colunas que o permitem.
- Para adicionar explicitamente um valor nulo a uma coluna que o permita, use a palavra-chave NULL na lista VALUES.

Inserindo Linhas com Valores Nulos

- Para especificar strings vazias e/ou datas ausentes, use aspas simples vazias (sem espaços entre elas: '') para os dados que estão faltando.

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number,
 hire_date, job_id, salary)
VALUES
  (302,'Grigorz','Polanski', 'gpolanski', '', '15-Jun-2015',
 'IT_PROG',4200);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
302	Grigorz	Polanski	gpolanski	-	15/Jun/2015	IT_PROG	4200

COMM_PCT	MGR_ID	DEPT_ID	BONUS
-	-	-	-

.....

Inserindo Valores Especiais

- Valores especiais, como SYSDATE e USER, podem ser incluídos na lista VALUES de uma instrução INSERT.
- SYSDATE colocará a hora e data atuais em um coluna.
- USER inserirá o nome de usuário da sessão atual, que é OAE_PUBLIC_USER no Oracle Application Express.

Inserindo Valores Especiais

- Este exemplo adiciona USER como o sobrenome, e SYSDATE como a data de contratação.

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number, hire_date,
 job_id, salary)
VALUES
  (304,'Test',USER, 't_user', 4159982010, SYSDATE, 'ST_CLERK',2500);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
304	Test	APEX_PUBLIC_USER	t_user	4159982010	15-Jun-2015	ST_CLERK	2500

- .....

COMM_PCT	MGR_ID	DEPT_ID	BONUS
-	-	-	-

Inserindo Valores de Data Específicos

- O modelo de formato padrão dos tipos de dados de data é DD-Mês-AAAA.
- Nesse formato de data, está incluída também a hora padrão de meia-noite (00:00:00).
- Em uma seção anterior, aprendemos a usar a função TO_CHAR para converter uma data para uma string de caracteres quando queremos recuperar e exibir um valor de data em um formato não padrão.
- Eis um lembrete de TO_CHAR:

```
SELECT first_name, TO_CHAR(hire_date,'Month, fmdd, yyyy')
FROM employees
WHERE employee_id = 101;
```

FIRST_NAME	TO_CHAR(HIRE_DATE,'MONTH,FMDD,YYYY')
Neena	September, 21, 1989

Inserindo Valores de Data Específicos

- Da mesma forma, se quisermos inserir uma linha com um formato não padrão para uma coluna de datas, devemos usar a função TO_DATE para converter o valor de data (uma string de caracteres) em uma data.

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number, hire_date,
 job_id, salary)
VALUES
 (301,'Katie','Hernandez', 'khernandez','8586667641',
  TO_DATE('July 8, 2015', 'Month fmdd, yyyy'), 'MK_REP',4200);
```

Inserindo Valores de Data Específicos

- Um segundo exemplo de TO_DATE permite a inserção de uma hora específica do dia, substituindo a hora padrão de meia-noite.

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number, hire_date,
 job_id, salary)
VALUES
  (303,'Angelina','Wright', 'awright','4159982010',
 TO_DATE('July 10, 2015 17:20', 'Month fmdd, yyyy HH24:MI'),
 'MK_REP', 3600);
```

```
SELECT first_name, last_name,
 TO_CHAR(hire_date, 'dd-Mon-YYYY HH24:MI') As "Date and Time"
FROM copy_employees
WHERE employee_id = 303;
```

FIRST_NAME	LAST_NAME	Date and Time
Angelina	Wright	10-Jul-2015 17:20

Usando uma Subconsulta para Copiar Linhas

- Cada instrução INSERT que vimos até agora adiciona somente uma linha à tabela.
- Mas vamos supor que queremos copiar 100 linhas de uma tabela para outra.
- Não queremos precisar gravar e executar 100 instruções INSERT separadas, uma depois da outra.
- Isso levaria muito tempo.
- Felizmente, a linguagem SQL nos permite usar uma subconsulta dentro de uma instrução INSERT.

Usando uma Subconsulta para Copiar Linhas

- Todos os resultados da subconsulta são inseridos na tabela.
- Portanto, podemos copiar 100 (ou 1.000) linhas com uma subconsulta multilinha em INSERT.
- Como já é de se esperar, você não precisa de uma cláusula VALUES quando está usando uma subconsulta para copiar linhas porque os valores inseridos serão exatamente os mesmos que os retornados pela subconsulta.

Usando uma Subconsulta para Copiar Linhas

- No exemplo mostrado, uma nova tabela chamada SALES_REPS será preenchida com cópias de algumas das linhas e colunas da tabela EMPLOYEES.
- A cláusula WHERE selecionará os funcionários que têm IDs de cargo como '%REP%'.

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
  SELECT employee_id, last_name, salary, commission_pct
  FROM employees
  WHERE  job_id LIKE '%REP%';
```

Usando uma Subconsulta para Copiar Linhas

- O número de colunas e os tipos de dados na lista de colunas da cláusula INSERT devem ser equivalentes ao número de colunas e tipos de dados na subconsulta.
- A subconsulta não fica entre parênteses, como é feito com as subconsultas na cláusula WHERE de uma instrução SELECT.

Usando uma Subconsulta para Copiar Linhas

- Se quisermos copiar todos os dados (todas as linhas e colunas), a sintaxe é ainda mais simples.
- Para selecionar todas as linhas da tabela EMPLOYEES e inseri-las na tabela SALES_REPS, a instrução seria gravada assim:

```
INSERT INTO sales_reps
 SELECT *
 FROM employees;
```

- Mais uma vez, isso funcionará somente se ambas as tabelas tiverem o mesmo número de colunas com tipos de dados equivalentes, e eles estiverem na mesma ordem.

Terminologia

Estes são os principais termos usados nesta lição:

- INSERT INTO
- USER
- Transação
- Explícito

Resumo

Nesta lição, você deverá ter aprendido a:

- Explicar a importância de ser capaz de alterar os dados em um banco de dados
- Construir e executar instruções INSERT que inserem uma única linha usando uma cláusula VALUES
- Construir e executar instruções INSERT que usam valores especiais, nulos e de data
- Construir e executar instruções INSERT que copiam linhas de uma tabela para outra usando uma subconsulta


ORACLE
Academy