





Instrução SQL

SELECT



Objetivos

FIAP

- Ao concluir esta lição, você será capaz de:
 - Listar os recursos das instruções SQL `SELECT`
 - Executar uma instrução `SELECT` básica

Objetivos

Para extrair dados do banco de dados, é necessário usar a instrução SQL (Structured Query Language) `SELECT`. Talvez você precise restringir as colunas exibidas. Esta lição descreve todas as instruções SQL necessárias para executar essas ações. Você pode criar instruções `SELECT` para usar mais de uma vez.

Esta lição também aborda o ambiente *SQL Developer* no qual as instruções SQL são executadas.

Recursos de Instruções SQL SELECT

FIAP

Projeção

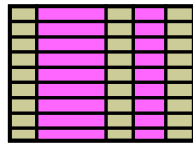


Tabela 1

Seleção

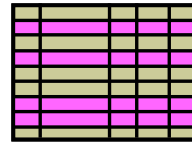


Tabela 1

Join

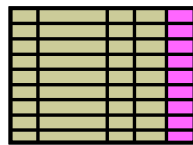


Tabela 1

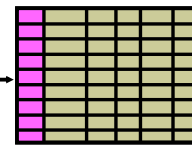


Tabela 2

Recursos de Instruções SQL SELECT

Uma instrução `SELECT` recupera informações do banco de dados. Com essa instrução, é possível usar os seguintes recursos:

Projeção: Escolha as colunas de uma tabela a serem retornadas por uma consulta. Escolha quantas colunas forem necessárias

Seleção: Escolha as linhas de uma tabela a serem retornadas por uma consulta. É possível usar vários critérios para restringir as linhas recuperadas.

Join: Une os dados armazenados em diferentes tabelas especificando o vínculo entre elas. As joins SQL são abordadas com mais detalhes em uma lição posterior.

Criando Instruções SQL

- As instruções SQL não fazem distinção entre maiúsculas e minúsculas.
- As instruções SQL podem ocupar uma ou mais linhas.
- Não é possível abreviar palavras-chave ou dividi-las em duas linhas.
- As cláusulas geralmente são colocadas em linhas separadas.
- Os recuos são usados para melhorar a legibilidade.
- O encerramento das instruções SQL com ponto-e-vírgula (;) é opcional.
- A utilização de ponto-e-vírgula será obrigatória se você executar várias instruções SQL.
- Você deverá encerrar cada instrução SQL com ponto-e-vírgula (;).

Criando Instruções SQL

Com estas diretrizes e regras simples, você pode criar instruções válidas de fácil leitura e edição:

As instruções SQL não fazem distinção entre maiúsculas e minúsculas (a menos que essa distinção seja indicada).

É possível informar instruções SQL em uma ou várias linhas.

Não é possível dividir palavras-chave em duas linhas ou abreviá-las.

As cláusulas normalmente são colocadas em linhas separadas por questões de legibilidade e facilidade de edição.

Devem ser usados recuos para tornar o código mais legível.

Em geral, as palavras-chave são informadas em maiúsculas; todas as outras palavras, como nomes de tabelas e colunas, são informadas em minúsculas.

Executando Instruções SQL

No iSQL*Plus, clique no botão Execute para executar os comandos na janela de edição.

No SQL*Plus, encerre a instrução SQL com ponto-e-vírgula e pressione a tecla Enter para executar o comando.

Instrução SELECT Básica

```
SELECT *|{[DISTINCT] column|expression [alias],...}  
FROM      table;
```

- SELECT identifica as colunas a serem exibidas
- FROM identifica a tabela contendo essas colunas

Instrução SELECT Básica

Em sua forma mais simples, uma instrução SELECT deve incluir:

Uma cláusula SELECT, que especifica as colunas a serem exibidas

Uma cláusula FROM, que identifica a tabela com as colunas listadas na cláusula SELECT

Na sintaxe:

SELECT

*

DISTINCT

column|expression

expressão

alias

selecionadas

FROM table

colunas

é uma lista de uma ou mais colunas

seleciona todas as colunas

suprime as colunas duplicadas

seleciona a coluna nomeada ou a

fornece cabeçalhos distintos às colunas

especifica a tabela que contém as

Observação: Neste curso, os termos *palavra-chave*, cláusula e instrução são usados da seguinte maneira:

Uma *palavra-chave* é um elemento individual de SQL.

Por exemplo, SELECT e FROM são palavras-chave.

Uma *cláusula* é uma parte de uma instrução SQL.

Por exemplo, SELECT employee_id, last_name, ... é uma cláusula.

Uma *instrução* é uma combinação de duas ou mais cláusulas.
Por exemplo, `SELECT * FROM employees` é uma instrução SQL.

Selecionando Todas as Colunas

```
SELECT *  
FROM departments;
```

Selecionando Colunas Específicas

```
SELECT department_id, location_id  
FROM departments;
```

Selecionando Todas as Colunas de Todas as Linhas

Você pode exibir todas as colunas de dados de uma tabela inserindo um asterisco (*) após a palavra-chave `SELECT`. No exemplo do slide, a tabela `departments` contém quatro colunas: `DEPARTMENT_ID`, `DEPARTMENT_NAME`, `MANAGER_ID` e `LOCATION_ID`. A tabela contém sete linhas, uma para cada departamento.

Você também pode exibir todas as colunas da tabela listando-as após a palavra-chave `SELECT`. Por exemplo, a instrução SQL a seguir (como o exemplo do slide) exibe todas as colunas e linhas da tabela `DEPARTMENTS`:

```
SELECT department_id, department_name, manager_id,  
location_id  
FROM departments;
```

Expressões Aritméticas

- Crie expressões com dados de número e data usando operadores aritméticos.

| Operador | Descrição |
|----------|-------------|
| + | Somar |
| - | Subtrair |
| * | Multiplicar |
| / | Dividir |

Expressões Aritméticas

Talvez você precise modificar a forma como os dados são exibidos, realizar cálculos ou examinar cenários hipotéticos. Todas essas ações são possíveis com o uso de expressões aritméticas. Uma expressão aritmética pode conter nomes de colunas, valores numéricos constantes e operadores aritméticos.

Operadores Aritméticos

O slide lista os operadores aritméticos disponíveis em SQL. Você pode usar operadores aritméticos em qualquer cláusula de uma instrução SQL (exceto na cláusula `FROM`).

Observação: Nos tipos de dados `DATE` e `TIMESTAMP`, só é possível usar os operadores de adição e subtração.

Usando Operadores Aritméticos

```
SELECT last_name, salary, salary + 300
FROM employees;
```

Precedência de Operadores

```
SELECT last_name, salary, 12*salary+100
FROM employees;
```

1

```
SELECT last_name, salary, 12*(salary+100)
FROM employees;
```

2

Usando Operadores Aritméticos

O exemplo do slide usa o operador de adição para calcular um aumento de salário de US\$ 300 para todos os funcionários. O slide também exibe uma coluna `SALARY+300` na saída.

Observe que a coluna `SALARY+300` calculada resultante não é uma nova coluna da tabela `EMPLOYEES`; ela existe apenas para fins de exibição. Por default, o nome de uma nova coluna origina-se do cálculo que a gerou – neste caso, `salary+300`.

Observação: O servidor Oracle ignora os espaços em branco antes e depois do operador aritmético.

Precedência de Operadores

Se uma expressão aritmética contiver mais de um operador, a multiplicação e a divisão serão avaliadas primeiro. Se os operadores de uma expressão tiverem a mesma prioridade, a avaliação será realizada da esquerda para a direita.

Você pode usar parênteses para impor a avaliação da expressão entre parênteses primeiro.

Regras de Precedência:

A multiplicação e a divisão ocorrem antes da adição e da subtração.

Os operadores com a mesma prioridade são avaliados da esquerda para a direita.

São usados parênteses para sobrepor a precedência default ou tornar

a instrução mais clara.

Definindo um Valor Nulo

- Um valor nulo não está disponível nem designado e não é conhecido ou aplicável.
- Um valor nulo é diferente de zero ou de um espaço em branco.

```
SELECT last_name, job_id, salary, commission_pct
FROM employees;
```

Valores Nulos em Expressões Aritméticas

- As expressões aritméticas que contêm um valor nulo são avaliadas como nulas.

```
SELECT last_name, 12*salary*commission_pct
FROM employees;
```

Valores Nulos

Se, em uma coluna específica, uma linha não contiver um valor de dados, o valor será *nulo* ou conterá um valor nulo.

Um valor nulo não está disponível nem designado e não é conhecido ou aplicável. Um valor nulo é diferente de zero ou de um espaço. Zero é um número e um espaço é um caractere.

As colunas de qualquer tipo de dados podem conter valores nulos. No entanto, algumas constraints (`NOT NULL` e `PRIMARY KEY`) impedem o uso de valores nulos em colunas.

Na coluna `COMMISSION_PCT` da tabela `EMPLOYEES`, observe que apenas um gerente de vendas ou um representante de vendas pode receber comissão. Os outros funcionários não têm direito a comissões. Um valor nulo representa essa situação.

Definindo um Apelido de Coluna

- Um apelido de coluna:
 - Renomeia um cabeçalho de coluna
 - É útil em cálculos
 - Aparece imediatamente após o nome da coluna (Também é possível incluir a palavra-chave opcional `AS` entre o nome e o apelido da coluna.
 - Requer aspas duplas quando contém espaços ou caracteres especiais, quando faz distinção entre maiúsculas e minúsculas

Apelidos de Colunas

Quando exibe o resultado de uma consulta, o *SQL Developer* geralmente usa o nome da coluna selecionada como seu cabeçalho. Como esse cabeçalho pode não ser descritivo, talvez seja difícil compreendê-lo. Para alterar um cabeçalho de coluna, use um apelido.

Especifique o apelido após a coluna na lista `SELECT` usando um espaço como separador. Por default, os cabeçalhos de apelidos aparecem em maiúsculas.

Se o apelido contiver espaços ou caracteres especiais (como `#` ou `$`), ou se fizer distinção entre maiúsculas e minúsculas, delimite-o por aspas duplas ("").

Usando Apelidos de Colunas

```
SELECT last_name AS name, commission_pct comm  
FROM employees;
```

```
SELECT last_name "Name" , salary*12 "Annual Salary"  
FROM employees;
```

Apelidos de Colunas (continuação)

O primeiro exemplo exibe os nomes e os percentuais de comissão de todos os funcionários. Observe que a palavra-chave opcional `AS` foi usada antes do apelido de coluna `name`. O resultado da consulta será o mesmo com ou sem a inclusão da palavra-chave `AS`. Observe também que, na instrução SQL, os apelidos das colunas, `name` e `comm`, estão em minúsculas, enquanto o resultado da consulta exibe os cabeçalhos das colunas em maiúsculas. Como mencionado em um slide anterior, os cabeçalhos das colunas aparecem em maiúsculas por default.

O segundo exemplo exibe os sobrenomes e os salários anuais de todos os funcionários. Como `Annual Salary` contém um espaço, ele foi delimitado por aspas duplas. Observe que o cabeçalho da coluna na saída é exatamente igual ao apelido da coluna.

Operador de Concatenação

- Um operador de concatenação:
 - Vincula colunas ou strings de caracteres a outras colunas
 - É representado por duas barras verticais (||)
 - Cria uma coluna resultante que é uma expressão de caracteres

```
SELECT      last_name||job_id AS "Employees"  
FROM        employees;
```

Operador de Concatenação

Você pode vincular uma coluna a outras colunas, expressões aritméticas ou valores de constantes para criar uma expressão de caracteres usando o *operador de concatenação* (||). As colunas nos dois lados do operador são combinadas para formar uma única coluna de saída.

No exemplo, LAST_NAME e JOB_ID são concatenadas e recebem o apelido Employees. Observe que o sobrenome do funcionário e o código do cargo são combinados para formar uma única coluna de saída.

A palavra-chave AS antes do apelido facilita a leitura da cláusula SELECT.

Valores Nulos com o Operador de Concatenação

Se você concatenar um valor nulo com uma string de caracteres, o resultado será uma string de caracteres. LAST_NAME || NULL resulta em LAST_NAME.

Strings de Caracteres Literais

- Um literal é um caractere, uma data ou um número incluído na instrução `SELECT`.
- É necessário delimitar os valores dos literais de caractere e data por aspas simples.
- Cada string de caracteres da saída corresponde a apenas uma linha retornada.

Strings de Caracteres Literais

Um literal é um caractere, uma data ou um número incluído na lista `SELECT` que não constitui um nome ou um apelido de coluna. Ele é impresso para cada linha retornada. É possível incluir strings de literais de texto em formato livre no resultado da consulta. Essas strings são tratadas como uma coluna na lista `SELECT`.

Os literais de caractere e data *devem* ser delimitados por aspas simples (' '); em literais de número, as aspas não são necessárias.

Linhas Duplicadas

- A exibição default de consultas mostra todas as linhas, inclusive as linhas duplicadas.

```
SELECT department_id  
FROM employees;
```

1

```
SELECT DISTINCT department_id  
FROM employees;
```

2

Linhas Duplicadas

A menos que seja feita outra especificação, o *SQL Developer* exibe os resultados de uma consulta sem eliminar as linhas duplicadas. O primeiro exemplo do slide exibe todos os números de departamentos da tabela `EMPLOYEES`. Observe que os números de departamentos são repetidos. Para eliminar linhas duplicadas do resultado, inclua a palavra-chave `DISTINCT` na cláusula `SELECT` logo após a palavra-chave `SELECT`. No segundo exemplo do slide, a tabela `EMPLOYEES` contém, na verdade, 20 linhas, mas existem apenas sete números de departamentos exclusivos na tabela.

Você pode especificar várias colunas após o qualificador `DISTINCT`. Esse qualificador afeta todas as colunas selecionadas e o resultado são todas as combinações distintas das colunas.

```
SELECT DISTINCT department_id, job_id  
FROM employees;
```

Sumário

- Nesta lição, você aprendeu a:
 - Criar uma instrução `SELECT` que:
 - Retorna todas as linhas e colunas de uma tabela
 - Retorna as colunas especificadas de uma tabela
 - Usa apelidos para exibir cabeçalhos de colunas mais descritivos

```
SELECT *|{[DISTINCT] column|expression [alias],...}  
FROM table;
```

Instrução `SELECT`

Nesta lição, você aprendeu a recuperar dados de uma tabela do banco de dados com a instrução `SELECT`.

```
SELECT *|{[DISTINCT] column [alias],...}  
FROM table;
```

Na sintaxe:

| | |
|--------------------------------|--|
| <code>SELECT</code> | é uma lista de uma ou mais colunas |
| <code>*</code> | seleciona todas as colunas |
| <code>DISTINCT</code> | suprime as colunas duplicadas |
| <code>column expression</code> | seleciona a coluna nomeada ou a expressão |
| <code>alias</code> | fornece cabeçalhos distintos às colunas selecionadas |
| <code>FROM table</code> | especifica a tabela que contém as colunas |

`iSQL*Plus`

SQL Developer é um ambiente de execução que permite enviar instruções SQL ao servidor de banco de dados, bem como editar e salvar essas instruções. É possível executar as instruções no prompt SQL ou em um arquivo de script.

Bibliografia Utilizada

FIAP

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 © 2004, 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.