



Programação de Banco de Dados com SQL

19-3

Revisão do Exame Final



Objetivos

Esta lição abrange os seguintes objetivos:

- Rever os pontos principais sobre manipulação de letras maiúsculas e minúsculas e de caracteres
- Rever funções gerais, de número, de data e de conversão
- Rever as condições de expressão
- Rever produto cartesiano e operações de junção
- Rever não-equijunções, junções externas, autojunções, junções cruzadas, junções naturais e cláusulas de junção
- Rever funções de grupo, sintaxe GROUP BY e cláusulas HAVING

Objetivos

Esta lição abrange os seguintes objetivos:

- Rever subconsultas de linha única e multilinha
- Rever subconsultas do tipo pair-wise e non-pair-wise
- Rever subconsultas correlacionadas
- Rever as instruções DML INSERT, UPDATE, DELETE e MERGE e instruções INSERT em várias tabelas
- Rever as instruções DDL CREATE, ALTER, RENAME, TRUNCATE, FLASHBACK TABLE, DROP e FLASHBACK QUERY
- Rever as instruções DCL CREATE e REVOKE para privilégios de objeto

Finalidade

- Uma revisão é a melhor maneira de se preparar para uma avaliação.
- A avaliação ajuda você a ver o quanto aprendeu e destaca as áreas nas quais talvez queira melhorar.
- A revisão dos tópicos aprendidos até agora vai ajudar você a fazer o seu melhor no exame final.

Revisão de Sintaxe

- Esta é uma revisão de sintaxe.
- Certifique-se de revisar também as regras relacionadas à sintaxe.
- Eles são abordadas durante o curso.

Manipulação de Letras Maiúsculas e Minúsculas e de Caracteres

- Letras Maiúsculas e Minúsculas

```
LOWER(nome da coluna|expressão)  
UPPER(nome da coluna|expressão)  
INITCAP(nome da coluna|expressão)
```

- Caracteres

```
CONCAT(nome da coluna|expressão, nome da coluna|expressão)  
SUBSTR(nome da coluna|expressão,n,m)  
LENGTH(nome da coluna|expressão)  
  
INSTR(nome da coluna|expressão, literal de string)  
LPAD (nome da coluna|expressão, n, literal de caractere)  
RPAD(nome da coluna|expressão, n, literal de caractere)  
TRIM ( [começo | final | ambos] car1 FROM car2)  
REPLACE (nome da coluna|expressão, string a ser substituída, string  
substituta)
```

Funções de Número

```
ROUND(coluna|expressão,n)  
TRUNC(coluna|expressão,n)  
MOD(coluna|expressão, coluna|expressão)
```


Funções de Data

```
ROUND(coluna | expressão, string)
```

```
TRUNC(coluna | expressão, string)
```

```
MONTHS_BETWEEN(coluna | expressão, coluna | expressão)
```

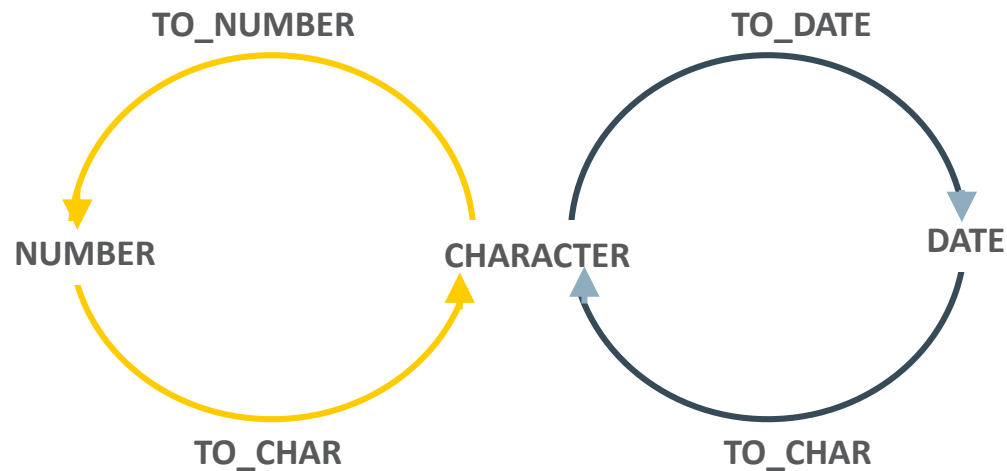
```
ADD_MONTHS(coluna | expressão, n)
```

```
NEXT_DAY(coluna | expressão, 'dia')
```

```
LAST_DAY(coluna | expressão)
```

Funções de Conversão

```
TO_CHAR(número, 'modelo de formato')  
TO_CHAR(data, 'modelo de formato')  
TO_NUMBER(string de caracteres, 'modelo de formato')  
TO_DATE(string de caracteres, 'modelo de formato')
```



Funções NULL

```
NVL(coluna|expressão, valor)
```

```
NVL2(coluna|expressão, coluna|expressão,  
      coluna|expressão)
```

```
NULLIF(coluna|expressão, coluna|expressão)
```

```
COALESCE(coluna|expressão, coluna|expressão,  
          coluna|expressão... coluna|expressão)
```

Expressões de Condição

- Específicas da Oracle

```
DECODE(coluna|expressão, pesquisa1, resultado1  
      [, pesquisa2, resultado2,...,]  
      [, padrão])
```

- ANSI

```
CASE expr WHEN expr1_comparação THEN retorno_expr1  
      [WHEN expr2_comparação THEN retorno_expr2  
      WHEN expr_comparação THEN retorno_expr  
      ELSE outra_expr]  
END
```

Sintaxe do Padrão ANSI SQL

- Junção Cruzada

```
SELECT last_name, department_name  
FROM employees CROSS JOIN departments;
```

- Junção Natural

```
SELECT employee_id, last_name, department_name  
FROM employees NATURAL JOIN departments;
```

- Join .. On

```
SELECT e.employee_id, e.last_name, e.salary, j.grade_level  
FROM employees e JOIN job_grades j  
ON (e.salary BETWEEN j.lowest_sal AND j.highest_sal);
```

Sintaxe do Padrão ANSI SQL

- Join .. Using

```
SELECT employee_id, last_name, department_name  
FROM employees JOIN departments  
USING (department_id);
```

- Join .. On

```
SELECT e.employee_id, e.last_name, d.department_id, d.location_id  
FROM employees e JOIN departments d  
ON (e.department_id = d.department_id);
```

Sintaxe do Padrão ANSI SQL

Junções Externas

- Junção Externa Direita

```
SELECT e.employee_id, e.last_name,  
       e.department_id, d.department_name  
FROM employees e RIGHT OUTER JOIN departments d  
ON (e.department_id = d.department_id);
```

- Junção Externa Esquerda

```
SELECT e.employee_id, e.last_name,  
       e.department_id, d.department_name  
FROM employees e LEFT OUTER JOIN departments d  
ON (e.department_id = d.department_id);
```

Sintaxe do Padrão ANSI SQL

Junções Externas

- Junção Externa Completa (Sem junção específica comparável do Oracle)

```
SELECT e.employee_id, e.last_name,  
       e.department_id, d.department_name  
FROM employees e FULL OUTER JOIN departments d  
ON (e.department_id = d.department_id);
```


Funções de grupo, sintaxe GROUP BY e cláusulas HAVING

```
AVG (coluna | expressão)
COUNT (coluna | expressão)
MIN (coluna | expressão)
MAX (coluna | expressão)
SUM (coluna | expressão)
VARIANCE (coluna | expressão)
STDDEV (coluna | expressão)
```

```
SELECT coluna1, AVG (coluna | expressão)
FROM tabela 1
GROUP BY (ROLLUP | CUBE) (coluna1 | GROUPING SETS)
HAVING AVG (coluna | expressão)
```

Subconsultas de Linha Única e Multilinha

```
SELECT coluna1..  
FROM tabela 1  
WHERE coluna2 = (SELECT coluna2  
                  FROM tabela 1  
                  WHERE coluna 3 = expressão)
```

- Operadores de linha única: =, >, <, >=, <=, <>
- Operadores multilinha: IN, ANY, ALL

Subconsultas do Tipo Pairwise e Non-Pairwise

- Pairwise

```
SELECT coluna1..  
FROM tabela 1  
WHERE (coluna2, coluna3) = (SELECT coluna2, coluna3  
                             FROM tabela 1  
                             WHERE coluna 4 = expressão);
```

- Non-pairwise

```
SELECT coluna1..  
FROM tabela 1  
WHERE coluna2 = (SELECT coluna2  
                 FROM tabela 1  
                 WHERE coluna 4 = expressão)  
AND   coluna3 = (SELECT coluna3  
                 FROM tabela 2  
                 WHERE coluna 4 = expressão);
```

Subconsultas Correlacionadas

```
SELECT o.coluna1..  
FROM tabela_1 o  
WHERE o.coluna2 = (SELECT i.coluna2  
                   FROM tabela_2 i  
                   WHERE i.coluna1 = o.coluna1)
```

Inserindo, Atualizando e Excluindo Dados

- Inserção Explícita

```
INSERT INTO tabela (coluna1, coluna2...)  
VALUES (valor1, valor2...) ;
```

- Inserção Implícita

```
INSERT INTO tabela  
VALUES (valor1, valor2, valor3, valor4);
```

```
UPDATE tabela1  
SET coluna1 = valor1,  
    coluna2 = valor2...  
WHERE coluna1 = valor;
```

```
DELETE FROM tabela1  
WHERE coluna1 = valor;
```

Inserindo, Atualizando e Excluindo Dados

```
UPDATE tabela1  
SET coluna1 = valor1,  
    coluna2 = valor2...  
WHERE coluna1 = valor;
```

```
DELETE FROM tabela1  
WHERE coluna1 = valor;
```

Inserindo, Atualizando e Excluindo Dados

```
cláusula_insert_condicional  
[ ALL | FIRST ]  
WHEN condição THEN  
    cláusula_insert_into [ cláusula_values ]  
WHEN condição THEN  
    cláusula_insert_into [ cláusula_values ]  
ELSE cláusula_insert_into [ cláusula_values ]
```

Valores Padrão

```
CREATE TABLE tabela1 (  
  coluna1          DATE DEFAULT SYSDATE,...)  
  
INSERT INTO tabela1  
  (coluna1,...)  
VALUES  
  (DEFAULT,...);
```


A Instrução Merge

Instrução INSERT em Várias Tabelas

```
MERGE INTO tabela-destino USING tabela-origem  
ON condição-equivalência  
WHEN MATCHED THEN UPDATE  
SET .....  
WHEN NOT MATCHED THEN INSERT  
VALUES (.....);
```

Criando Tabelas

```
CREATE TABLE tabela  
(coluna tipo de dados [DEFAULT expressão],  
coluna tipo de dados [DEFAULT expressão],  
.....[ ] );
```

```
CREATE TABLE nomedatabela  
[(coluna, coluna, ...)]  
AS subconsulta;
```

Especificando Tipos de Dados

```
NUMBER(p,s)  
CHAR  
VARCHAR2(n)  
DATE  
TIMESTAMP  
TIMESTAMP WITH TIMEZONE  
TIMESTAMP WITH LOCAL TIME ZONE  
INTERVAL YEAR TO MONTH  
INTERVAL DAY TO SECOND  
CLOB  
BLOB  
RAW
```

Modificando uma Tabela

```
ALTER TABLE nomedatabela  
ADD (nome_coluna tipo de dados [DEFAULT expressão]...);
```

```
ALTER TABLE nomedatabela MODIFY (nome_coluna VARCHAR2(30));
```

```
ALTER TABLE nomedatabela DROP COLUMN nome da coluna;
```

```
ALTER TABLE nomedatabela SET UNUSED (nome da coluna);
```

```
ALTER TABLE nomedatabela DROP UNUSED COLUMNS;
```

Modificando uma Tabela

- Instrução INSERT em Várias Tabelas

```
DROP TABLE nomedatabela;
```

```
FLASHBACK TABLE nomedatabela TO BEFORE DROP;
```

```
SELECT * FROM user_recyclebin;
```

```
SELECT versions_starttime "START_DATE",  
       versions_endtime   "END_DATE",  
       coluna, coluna.....  
FROM   tabela  
       VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE  
WHERE  coluna = valor;
```

Constraints no Nível da Coluna

```
CREATE TABLE tabela
(col1 tipo de dados CONSTRAINT tab_col1_pk PRIMARY KEY,
 col2 tipo de dados CONSTRAINT tab_col2_nn NOT NULL,
 col3 tipo de dados CONSTRAINT tab_col3_uk UNIQUE,
 col4 tipo de dados CONSTRAINT tab_col4_ck CHECK (col4 > valor),
 col5 tipo de dados CONSTRAINT tab_col5 REFERENCES tabela2 (col1));
```

Constraints no Nível da Tabela

```
CREATE TABLE tabela
(col1 tipo de dados,
 col2 tipo de dados,
 col3 tipo de dados,
 col4 tipo de dados,
 col5 tipo de dados,
 CONSTRAINT tab_col1_pk PRIMARY(col1),
 CONSTRAINT tab_col3_uk UNIQUE(col2),
 CONSTRAINT tab_col4_ck CHECK (col4 > valor),
 CONSTRAINT tab1_col5_fk FOREIGN KEY (col5) REFERENCES tabela2 (col1));
```

Criando e Gerenciando Views

```
CREATE [OR REPLACE] [FORCE| NOFORCE] VIEW view [(alias [, alias]...)]  
AS subconsulta  
[WITH CHECK OPTION [CONSTRAINT constraint]]  
[WITH READ ONLY [CONSTRAINT constraint]];
```

```
DROP VIEW nomedaview;
```


Análise Top-N

```
SELECT ROWNUM as RANK, col1, col2
FROM (SELECT col1, col2 FROM tabela1
ORDER BY col1)
WHERE ROWNUM <= n;
```

Views em Linha

```
SELECT t1.col1, t2.col2...  
FROM tabela 1 t1, (SELECT col1, col2..  
                   FROM tabela2  
                   WHERE ...) t2  
WHERE .....;
```

Criando Sequências

```
CREATE SEQUENCE sequência  
  [INCREMENT BY n]  
  [START WITH n]  
  [{MAXVALUE n | NOMAXVALUE}]  
  [{MINVALUE n | NOMINVALUE}]  
  [{CYCLE | NOCYCLE}]  
  [{CACHE n | NOCACHE}];
```

```
DROP SEQUENCE nome_sequência;
```

Criando Índices e Sinônimos

```
CREATE INDEX nome_índice  
ON  nome_tabela( coluna...,coluna);
```

```
DROP INDEX nome_índice;
```

```
CREATE [PUBLIC] SYNONYM sinônimo  
FOR objeto;
```

```
DROP [PUBLIC] SYNONYM nome_do_sinônimo
```

Criando e Revogando Privilégios de Objeto

```
CREATE USER usuário  
IDENTIFIED BY senha;
```

```
GRANT privilégio [, privilégio...]  
TO usuário [, usuário| atribuição, PUBLIC...];
```

```
ALTER USER usuário  
IDENTIFIED BY senha;
```

Criando e Revogando Privilégios de Objeto

```
CREATE ROLE nome_atribuição;
```

```
GRANT priv_objeto [(lista_colunas)]  
ON nome_objeto  
TO {usuário|atribuição|PUBLIC}  
[WITH GRANT OPTION];
```

```
REVOKE {privilégio [, privilégio...]|ALL}  
ON objeto  
FROM {usuário[, usuário...]|atribuição|PUBLIC}  
[CASCADE CONSTRAINTS];
```

Resumo

Nesta lição, você deverá ter revisado:

- Os pontos principais sobre manipulação de letras maiúsculas e minúsculas e de caracteres
- Funções gerais, de número, de data e de conversão
- Expressões de condição
- Produto cartesiano e operações de junção
- Não-equijunções, junções externas, autojunções, junções cruzadas, junções naturais e cláusulas de junção
- Funções de grupo, sintaxe GROUP BY e cláusulas HAVING

Resumo

Nesta lição, você deverá ter revisado:

- Subconsultas de linha única e multilinha
- Subconsultas do tipo pair-wise e non-pair-wise
- Subconsultas correlacionadas
- Instruções DML INSERT, UPDATE, DELETE e MERGE e instruções INSERT em várias tabelas
- Instruções DDL FLASHBACK TABLE, DROP e FLASHBACK QUERY

