

A Linguagem SQL

Introdução a Banco de Dados – DCC 011

Introdução

- Originalmente proposta para o System R desenvolvido nos laboratórios da IBM na década de 70 → SEQUEL (Structured English QUEry Language)
- Objeto de um esforço de padronização coordenado pelo ANSI/ISO:
 - SQL1 (SQL-86)
 - SQL2 (SQL-92)
 - SQL3 (SQL-99)

Introdução a Banco de Dados – DCC 011

Introdução

- SQL = LDD + LMD + LCD
- Principais comandos:
 - LDD:
 - CREATE SCHEMA / TABLE / VIEW
 - DROP SCHEMA / TABLE / VIEW
 - ALTER TABLE
 - LMD:
 - SELECT, INSERT, UPDATE, DELETE
 - LCD:
 - GRANT, REVOKE
- Do ponto de vista de consulta, SQL é uma linguagem relacionalmente completa, ou seja, possui o mesmo poder de expressão da álgebra e do cálculo relacional

Introdução a Banco de Dados – DCC 011

Definição de Dados em SQL

- Comando CREATE SCHEMA
 - **CREATE SCHEMA** COMPANY **AUTORIZATION** JS;
- Comando CREATE TABLE
 - **CREATE TABLE** <nome da tabela>
(<definições de colunas>
 <definição da chave primária>
 <definições de chaves alternativas>
 <definições de chaves estrangeiras>);
 - **CREATE TABLE** <esquema>.<nome da tabela>

Introdução a Banco de Dados – DCC 011

Definição de Dados em SQL

Exemplo de um comando CREATE TABLE

```
CREATE TABLE EMPREGADO
(FNOME          VARCHAR(15) NOT NULL,
 MINICIAL       CHAR,
 UNAME          VARCHAR(15) NOT NULL,
 SSN            CHAR(9) NOT NULL,
 ...
 SUPERSSN       CHAR(9),
 DNO            INT NOT NULL,
 PRIMARY KEY (SSN),
 FOREIGN KEY (SUPERSSN) REFERENCES EMPREGADO (SSN)
 ON DELETE SET NULL,
 FOREIGN KEY (DNO) REFERENCES DEPARTAMENTO(DNUMERO));
```

Introdução a Banco de Dados – DCC 011

Definição de Dados em SQL

- Comando CREATE DOMAIN
CREATE DOMAIN TIPO_SSN **AS** CHAR(9);
CREATE TABLE EMPREGADO
(...
SSN TIPO_SSN **NOT NULL**,
...);
- Especificação de restrições sobre atributos
DNUMERO INT **NOT NULL**
CHECK (DNUMERO > 0 **AND** DNUMERO < 21)
Alternativamente:
CREATE DOMAIN D_NUMERO **AS** INTEGER
CHECK (D_NUMERO > 0 **AND** D_NUMERO < 21);
CREATE TABLE DEPARTAMENTO
(DNOME VARCHAR(15) **NOT NULL**,
DNUMERO D_NUMERO **NOT NULL**,
...);

Introdução a Banco de Dados – DCC 011

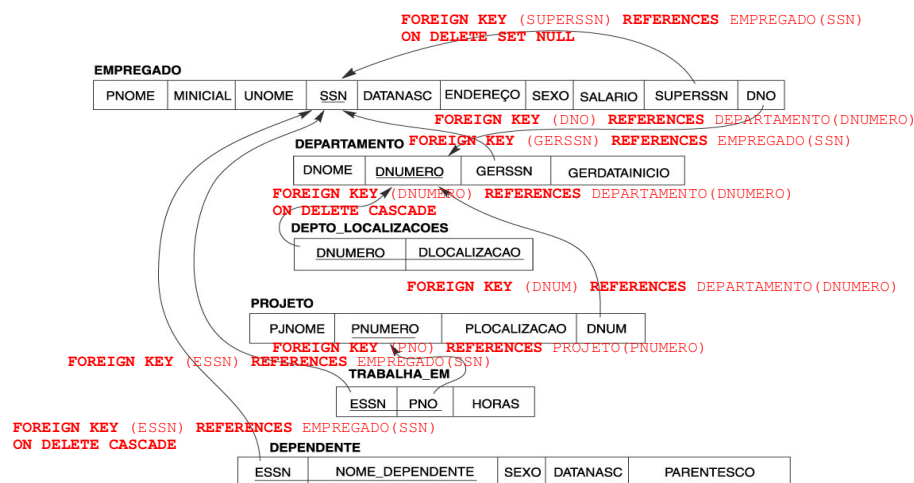
Definição de Dados em SQL

- Opções de remoção (cláusula ON DELETE):
 - CASCADE (propagação)
 - SET NULL (substituição por nulos)
 - SET DEFAULT (substituição por um valor default)
 - Opção default: **bloqueio** (RESTRICT)
- As mesmas opções se aplicam à cláusula ON UPDATE

Entretanto esta cláusula **NÃO** deve ser utilizada para se evitar que as chaves primárias sejam modificadas

Introdução a Banco de Dados – DCC 011

Definição de Dados em SQL Restrição de Integridade Referencial



Introdução a Banco de Dados – DCC 011

Definição de Dados em SQL

Restrição de Integridade Referencial

```
CREATE TABLE EMPREGADO
(PNOME VARCHAR(15) NOT NULL,
MINICIAL CHAR,
UNOME VARCHAR(15) NOT NULL,
SSN CHAR(9) NOT NULL,
DATANASC DATE,
ENDERECO VARCHAR(30),
SEXO CHAR,
SALARIO DECIMAL(10,2),
SUPERSSN CHAR(9),
DNO INT NOT NULL,
PRIMARY KEY (SSN),
FOREIGN KEY (SUPERSSN) REFERENCES EMPREGADO (SSN)
ON DELETE SETNULL,
FOREIGN KEY (DNO) REFERENCES DEPARTAMENTO (DNUMERO));

CREATE TABLE DEPARTAMENTO
(DNOME VARCHAR(15) NOT NULL,
DNUMERO INT NOT NULL,
GERSSN CHAR(9) NOT NULL,
GERDATAINICIO DATE,
PRIMARY KEY (DNUMERO),
UNIQUE (DNOME),
FOREIGN KEY (MGRSSN) REFERENCES EMPREGADO (SSN));

CREATE TABLE DPTO_LOCALIZACOES
(DNUM INT NOT NULL,
DLOCALIZACAO VARCHAR(15) NOT NULL,
PRIMARY KEY (DNUM, DLOCALIZACAO),
FOREIGN KEY (DNUM) REFERENCES DEPARTAMENTO (DNUMERO)
ON DELETE CASCADE);

CREATE TABLE PROJETO
(PJNOME VARCHAR(15) NOT NULL,
PNUMERO INT NOT NULL,
PLOCALIZACAO CHAR(9) NOT NULL,
DNUM INT NOT NULL,
PRIMARY KEY (PNUMERO),
UNIQUE (PJNOME),
FOREIGN KEY (DNUM) REFERENCES DEPARTAMENTO (DNUMERO));

CREATE TABLE TRABALHA_EM
(ESSN CHAR(9) NOT NULL,
PNO INT NOT NULL,
HORAS DECIMAL(3,1) NOT NULL,
DNUM INT NOT NULL,
PRIMARY KEY (ESSN, PNO),
FOREIGN KEY (ESSN) REFERENCES EMPREGADO (SSN);
FOREIGN KEY (PNO) REFERENCES PROJETO (PNUMERO));

CREATE TABLE DEPENDENTE
(ESSN CHAR(9) NOT NULL,
NOME_DEPENDENTE VARCHAR(15) NOT NULL,
SEXO CHAR NOT NULL,
PRIMARY KEY (ESSN, NOME_DEPENDENTE),
FOREIGN KEY (ESSN) REFERENCES EMPREGADO (SSN)
ON DELETE CASCADE;
```

Introdução a Banco de Dados – DCC 011

Definição de Dados em em SQL

Restrição de Integridade Referencial

CREATE TABLE EMPREGADO

```
(...
SSN CHAR(9) NOT NULL,
...
SUPERSSN CHAR(9),
DNO INT NOT NULL DEFAULT 1,
PRIMARY KEY (SSN),
FOREIGN KEY (SUPERSSN) REFERENCES
EMPREGADO (SSN) ON DELETE SET NULL,
FOREIGN KEY (DNO) REFERENCES
DEPARTAMENTO (DNUMERO) ON DELETE SET DEFAULT
ON UPDATE CASCADE);
```

Obs: sempre que possível deve-se evitar a opção **ON UPDATE CASCADE** (Por quê?)

Introdução a Banco de Dados – DCC 011

Definição de Dados em SQL

Restrição de Integridade Referencial

```
CREATE TABLE EMPREGADO
(
  ...,
  DNO INT NOT NULL DEFAULT 1,
CONSTRAINT EMPPK
  PRIMARY KEY (SSN),
CONSTRAINT EMPSUPERFK
  FOREIGN KEY (SUPERSSN) REFERENCES
  EMPREGADO (SSN) ON DELETE SET NULL,
CONSTRAINT EMPDEPTFK
  FOREIGN KEY (DNO) REFERENCES
  DEPARTAMENTO (DNUMERO) ON DELETE SET DEFAULT
  ON UPDATE CASCADE);
```

Introdução a Banco de Dados – DCC 011

Definição de Dados em SQL

Comandos para Modificação de Esquema

- Comandos DROP SCHEMA e DROP TABLE
 - **DROP SCHEMA** EMPRESA **CASCADE**;
 - **DROP TABLE** DEPENDENTE **CASCADE**;
- Comando ALTER TABLE
 - **ALTER TABLE** EMPRESA.EMPREGADO **ADD** FUNCAO VARCHAR(12);
 - **ALTER TABLE** EMPRESA.EMPREGADO **DROP** ENDERECO **CASCADE**;

Introdução a Banco de Dados – DCC 011

Consultas Básicas em SQL

- Formato básico do comando SELECT:

```
SELECT <lista de atributos>  
FROM <lista de tabelas>  
WHERE <condição>;
```

- Exemplo:

```
SELECT DATANASC, ENDERECO  
FROM EMPREGADO  
WHERE PNAME='John' AND  
        MINICIAL='B' AND  
        UNOME='Smith';
```

Introdução a Banco de Dados – DCC 011

Consultas Básicas em SQL

- SELECT** PNAME, UNOME, ENDERECO
FROM EMPREGADO, DEPARTAMENTO
WHERE DNAME='Pesquisa' **AND** DNO=DNUMBER;



- SELECT** PNUMBER, DNUM, UNOME, ENDERECO, DATANASC
FROM PROJETO, DEPARTAMENTO, EMPREGADO
WHERE PLOCALIZACAO='Stafford' **AND**
 DNUM=DNUMBER **AND** GERSSN=SSN;

Introdução a Banco de Dados – DCC 011

Consultas Básicas em SQL

- Atributos ambíguos e pseudônimos (*alias*)

```
SELECT D.NOME, D.LOCALIZACAO  
FROM DEPARTAMENTO, DEPTO_LOCALIZACOES  
WHERE DEPARTAMENTO.DNUMERO =  
        DEPTO_LOCALIZACOES.DNUMERO;
```

```
SELECT E.PNOME, E.UNAME, S.PNOME, S.UNAME  
FROM EMPREGADO AS E, EMPREGADO AS S  
WHERE E.SUPERSSN=S.SSN;
```

Introdução a Banco de Dados – DCC 011

Consultas Básicas em SQL

- Consultas sem a cláusula WHERE

```
SELECT SSN, UNOME, SALARIO  
FROM EMPLOYEE;
```

```
SELECT UNAME, D.NOME  
FROM EMPREGADO, DEPARTAMENTO  
WHERE DNO=DNUMERO;
```

Atenção: A ausência da cláusula WHERE torna a consulta acima um produto cartesiano das tabelas EMPREGADO e DEPARTAMENTO.

Introdução a Banco de Dados – DCC 011

Consultas Básicas em SQL

- Manipulando tabelas como conjuntos

```
SELECT SALARIO  
FROM EMPREGADO;
```

Não elimina linhas (tuplas) duplicatas

```
SELECT DISTINCT SALARIO  
FROM EMPREGADO;
```

```
(SELECT DISTINCT PNUMERO  
FROM PROJETO, DEPARTAMENTO, EMPREGADO  
WHERE DNUM=DNUMERO AND GERSSN=SSN AND  
UNOME='Smith')
```

UNION

```
(SELECT DISTINCT PNUMERO  
FROM PROJETO, TRABALHA_EM, EMPREGADO  
WHERE PNUMERO=PNO AND ESSN=SSN AND  
UNOME='Smith');
```

Introdução a Banco de Dados – DCC 011

Consultas Complexas em SQL

- Consultas aninhadas

```
SELECT PNUMERO, UNOME, ENDEREÇO  
FROM EMPREGADO  
WHERE DNO IN (SELECT DNUMERO  
                  FROM DEPARTAMENTO  
                  WHERE DNOME='Pesquisa');
```

é equivalente à consulta

```
SELECT PNUMERO, UNOME, ENDEREÇO  
FROM EMPREGADO, DEPARTAMENTO  
WHERE DNO=DNUMERO AND DNOME='Pesquisa';
```

Introdução a Banco de Dados – DCC 011

Consultas Complexas em SQL

- Comparação de conjuntos

```
SELECT DISTINCT PNUMERO
FROM PROJETO
WHERE PNUMERO IN (SELECT PNUMERO
                    FROM PROJETO, DEPARTAMENTO,
                    EMPREGADO
                    WHERE DNUM =DNUMERO AND
                    GERSSN=SSN AND
                    UNOME='Smith')

OR
PNUMERO IN (SELECT PNO
            FROM TRABALHA_EM, EMPREGADO
            WHERE ESSN=SSN AND
            UNOME='Smith');
```

Introdução a Banco de Dados – DCC 011

Consultas Complexas em SQL

- Comparação de conjuntos

```
SELECT DISTINCT ESSN
FROM TRABALHA_EM
WHERE (PNO, HORAS) IN (SELECT PNO, HORAS
                        FROM TRABALHA_EM
                        WHERE ESSN='123456789');

SELECT UNOME, PNO
FROM EMPREGADO
WHERE SALARIO > ALL (SELECT SALARIO
                     FROM EMPREGADO
                     WHERE DNO=5);
```

Introdução a Banco de Dados – DCC 011

Consultas Complexas em SQL

- Uso da função EXISTS

```
SELECT E.PNOME, E.UNOME  
FROM EMPREGADO AS E  
WHERE EXISTS (SELECT *  
                FROM DEPENDENTE  
                WHERE E.SSN=ESSN AND  
                  E.SEXO=SEXO AND  
                  E.PNOME=DEPENDENTE_NOME);
```

```
SELECT PNOME, UNOME  
FROM EMPREGADO  
WHERE NOT EXISTS (SELECT *  
                  FROM DEPENDENTE  
                  WHERE SSN=ESSN);
```

Introdução a Banco de Dados – DCC 011

Consultas Complexas em SQL

- Uso do operador CONTAINS

```
SELECT PNOME, UNOME  
FROM EMPREGADO  
WHERE ((SELECT PNO  
        FROM TRABALHA_EM  
        WHERE SSN=ESSN)  
        CONTAINS  
        (SELECT PNUMERO  
        FROM PROJETO  
        WHERE DNUM=5));
```

Esta consulta SQL corresponde a uma operação de divisão na álgebra relacional (verifique a Consulta 3 da Seção 6.5)

Introdução a Banco de Dados – DCC 011

Facilidades Adicionais

- Uso do operador JOIN

```
SELECT PNAME, UNOME, ENDereco  
FROM (EMPREGADO JOIN DEPARTAMENTO  
      ON DNO=DNUMERO)  
WHERE DNAME='Pesquisa';
```

```
SELECT DNAME, DLOCALIZACAO  
FROM (DEPARTAMENTO NATURAL JOIN  
      DEPTO_LOCALIZACOES);
```

```
SELECT PNAME, UNOME, DEPENDENTE_NOME  
FROM (EMPREGADO LEFT OUTER JOIN DEPENDENTE  
      ON SSN=ESSN);
```

Introdução a Banco de Dados – DCC 011

Facilidades Adicionais

- Funções de agregação

```
SELECT SUM(SALARIO), MAX(SALARIO), MIN(SALARIO),  
      AVG(SALARIO)  
FROM EMPREGADO;
```

```
SELECT SUM(SALARIO), MAX(SALARIO), MIN(SALARIO),  
      AVG(SALARIO)  
FROM EMPREGADO, DEPARTAMENTO  
WHERE DNO=DNUMERO AND DNAME='Pesquisa';
```

```
SELECT COUNT(*)  
FROM EMPREGADO, DEPARTAMENTO  
WHERE DNO=DNUMERO AND DNAME='Pesquisa';
```

Introdução a Banco de Dados – DCC 011

Facilidades Adicionais

■ Agrupamento

```
SELECT DNO, COUNT(*), AVG(SALARIO)  
FROM EMPREGADO  
GROUP BY DNO;
```

GROUP BY DNO



EMPREGADO

PNOME	MINIAIS	UNOME	SSN	...	SALARIO	SUPERSSN	DNO
John	B	Smith	123456789	...	30000	333445555	5
Franklin	T	Wong	333445555		40000	888666555	5
Alicia	J	Zelaya	999887777		25000	987654321	4
Jennifer	S	Wallace	987654321		43000	888666555	4
Ramesh	K	Narayan	666884444		38000	333445555	5
Joyce	A	English	453453453		25000	333445555	5
Ahmad	V	Jabbar	987987987		25000	987654321	4
James	E	Borg	888665555		55000	nulo	1

Introdução a Banco de Dados – DCC 011

Facilidades Adicionais

■ Agrupamento

```
SELECT DNO, COUNT(*), AVG(SALARIO)  
FROM EMPREGADO  
GROUP BY DNO;
```

GROUP BY DNO



EMPREGADO

PNOME	MINIAIS	UNOME	SSN	...	SALARIO	SUPERSSN	DNO
John	B	Smith	123456789	...	30000	333445555	5
Franklin	T	Wong	333445555		40000	888666555	5
Ramesh	K	Narayan	666884444		38000	333445555	5
Joyce	A	English	453453453		25000	333445555	5
Alicia	J	Zelaya	999887777		25000	987654321	4
Jennifer	S	Wallace	987654321		43000	888666555	4
Ahmad	V	Jabbar	987987987		25000	987654321	4
James	E	Borg	888665555		55000	nulo	1

SELECT DNO, COUNT(*), AVG(SALARIO)

DNO	COUNT(*)	AVG(SALARIO)
5	4	30000
4	3	40000
1	1	38000

Introdução a Banco de Dados – DCC 011

Facilidades Adicionais

- Agrupamento com a cláusula HAVING

```
SELECT PNUMERO, PJNOME, COUNT(*)  
FROM PROJETO, TRABALHA_EM  
WHERE PNUMERO=PNO  
GROUP BY PNUMERO, PJNOME  
HAVING COUNT(*) > 2;
```

Introdução a Banco de Dados – DCC 011

Facilidades Adicionais

```
SELECT ...  
FROM PROJETO, TRABALHA_EM  
WHERE PNUMERO=PNO
```

PROJETO

PJNOME	PNUMERO	PLOCALIZACAO	DNUM
ProdutoX	1	Bellaire	5
ProdutoY	2	Sugarland	5
ProdutoZ	3	Houston	5
Automacao	10	Houston	4
Reorganizacao	20	Stafford	1
NovosBeneficios	30	Houston	4

TRABALHA_EM

SSN	PNO	HORAS
123456789	1	32,5
453453453	1	20,0
123456789	2	7,5
453453453	2	20,0
333445555	2	10,0
666884444	3	40,0
333445555	3	10,0
333445555	10	10,0
999887777	10	10,0
987987987	10	35,0
333445555	20	10,0
987654321	20	15,0
888666555	20	nulo
987987987	30	5,0
987654321	30	20,0
999887777	30	30,0

Introdução a Banco de Dados – DCC 011

Facilidades Adicionais

GROUP BY PNUMERO, PJNOME
HAVING COUNT(*) > 2

PJNOME	PNUMERO	...	SSN	PNO	HORAS
ProdutoX	1		123456789	1	32,5
ProdutoX	1		453453453	1	20,0
ProdutoY	2		123456789	2	7,5
ProdutoY	2		453453453	2	20,0
ProdutoY	2		333445555	2	10,0
ProdutoZ	3		666884444	3	40,0
ProdutoZ	3		333445555	3	10,0
Automacao	10	...	333445555	10	10,0
Automacao	10		999887777	10	10,0
Automacao	10		987987987	10	35,0
Reorganizacao	20		333445555	20	10,0
Reorganizacao	20		987654321	20	15,0
Reorganizacao	20		888666555	20	nulo
NovosBeneficios	30		987987987	30	5,0
NovosBeneficios	30		987654321	30	20,0
NovosBeneficios	30		999887777	30	30,0

Introdução a Banco de Dados – DCC 011

Facilidades Adicionais

SELECT PNUMERO, PJNOME, COUNT(*)

PJNOME	PNUMERO	...	SSN	PNO	HORAS
ProdutoY	2		123456789	2	7,5
ProdutoY	2		453453453	2	20,0
ProdutoY	2		333445555	2	10,0
Automacao	10	...	333445555	10	10,0
Automacao	10		999887777	10	10,0
Automacao	10		987987987	10	35,0
Reorganizacao	20		333445555	20	10,0
Reorganizacao	20		987654321	20	15,0
Reorganizacao	20		888666555	20	nulo
NovosBeneficios	30		987987987	30	5,0
NovosBeneficios	30		987654321	30	20,0
NovosBeneficios	30		999887777	30	30,0

PNUMERO	PJNOME	COUNT(*)
2	ProdutoY	3
10	Automacao	3
20	Reorganizacao	3
30	NovosBeneficios	3

Introdução a Banco de Dados – DCC 011

Atualizações em SQL

- Comando INSERT

```
INSERT INTO EMPREGADO  
VALUES ('Richard','K','Marini','653258653','1962-12-30',  
'98 Oak Forest, Katy, TX',37000,'987654321',4);
```

```
INSERT INTO EMPREGADO(PNOME, UNOME, SSN, DNO)  
VALUES ('Richard','Marini','653258653',4);
```

```
INSERT INTO EMPREGADO(PNOME, UNOME, SSN, DNO)  
SELECT * FROM ENTRADA;
```

Introdução a Banco de Dados – DCC 011

Atualizações em SQL

- Comando DELETE

```
DELETE FROM EMPREGADO  
WHERE UNOME='Brown';
```

```
DELETE FROM EMPREGADO  
WHERE DNO IN (SELECT DNUMERO  
              FROM DEPARTAMENTO  
              WHERE DNOME='Pesquisa');
```

```
DELETE FROM EMPREGADO;
```

Introdução a Banco de Dados – DCC 011

Atualizações em SQL

- Comando UPDATE

```
UPDATE PROJETO  
SET PLOCALIZACAO='Bellaire', DNUM=5  
WHERE PNUMERO=10;
```

```
UPDATE EMPREGADO  
SET SALARIO=SALARIO*1.1  
WHERE DNO IN (SELECT DNUMERO  
                FROM DEPARTAMENTO  
                WHERE DNAME='Pesquisa');
```

Introdução a Banco de Dados – DCC 011

Visões em SQL

- Uma visão em SQL é uma tabela derivada de tabelas base (definidas no esquema) de um banco de dados ou de visões previamente definidas
- Comando CREATE VIEW
 - **CREATE VIEW** <nome da visão> [(<atributos>)]
AS <comando SELECT>;
- Visões simplificam a interface com o usuário e constituem um mecanismo eficiente de segurança

Introdução a Banco de Dados – DCC 011

Visões em SQL

- Exemplos:
 - TRABALHA_EM1(PNOME, UNOME, PJNOME, HORAS)
CREATE VIEW TRABALHA_EM1
AS SELECT PNOME, UNOME, PJNAME, HORAS
FROM EMPREGADO, PROJETO, TRABALHA_EM
WHERE SSN=ESSN **AND** PNO=PNUMERO;
 - DEPT_INFO (DEPT_NOME, NO_EMPS, TOTAL_SAL)
CREATE VIEW DEPT_INFO (DEPT_NOME, NO_EMPS,
TOTAL_SAL)
AS SELECT DNAME, **COUNT**(*), **SUM**(SALARIO)
FROM DEPARTAMENTO, EMPREGADO
WHERE DNUMERO=DNO
GROUP BY DNAME;

Introdução a Banco de Dados – DCC 011

Visões em SQL

- Visões podem ser consultadas como qualquer outra tabela
SELECT LNAME, HORAS
FROM TRABALHA_EM1
WHERE PNOME='ProjetoX';
- Visões sempre refletem o estado do banco de dados → não são materializadas quando definidas

Introdução a Banco de Dados – DCC 011

Implementação de Visões

- Processamento de uma consulta sobre uma visão → depende da estratégia adotada para sua implementação
 - Materialização da visão (*view materialization*)
 - A consulta é executada diretamente sobre a visão materializada
 - Modificação da consulta (*query modification*)
 - A consulta é modificada de acordo com a definição da visão e então executada sobre o banco de dados

Introdução a Banco de Dados – DCC 011

Implementação de Visões

- Estratégia de modificação da consulta

```
SELECT UNOME, HORAS  
FROM TRABALHA_EM1  
WHERE PJNOME='ProjetoX';
```



```
SELECT UNOME, HORAS  
FROM EMPREGADO, PROJETO, TRABALHA_EM  
WHERE SSN=ESSN AND  
PNO=PNUMERO AND  
PJNOME='ProjetoX';
```

Introdução a Banco de Dados – DCC 011

Atualização de Visões

- Em geral, visões **não** podem ser atualizadas
- Por exemplo, seja a seguinte operação de atualização sobre a visão TRABALHA_EM1:

```
UPDATE TRABALHA_EM1  
SET PJNOME='ProjetoY'  
WHERE UNOME='Smith' AND PNOME='John' AND  
      PJNOME='ProjetoX';
```

Como propagar esta atualização para as tabelas base?

Introdução a Banco de Dados – DCC 011

Atualização de Visões

- Opções para atualização de TRABALHA_EM1

- Opção 1:

```
UPDATE TRABALHA_EM  
SET PNO=(SELECT PNUMERO FROM PROJETO  
          WHERE PJNOME='ProjetoY')  
WHERE ESSN=(SELECT SSN FROM EMPREGADO  
            WHERE UNOME='Smith' AND  
              PNOME='John') AND  
      PNO=(SELECT PNUMERO FROM PROJETO  
          WHERE PJNOME='ProjetoX');
```

- Opção 2 (menos elaborada):

```
UPDATE PROJETO  
SET PJNOME='ProjetoY'  
WHERE PJNAME='ProjetoX';
```

Introdução a Banco de Dados – DCC 011

Atualização de Visões

- Visões definidas sobre uma única tabela são atualizáveis se seus atributos incluem a chave primária (ou uma das chave alternativas) da tabela base
- Visões definidas sobre múltiplas relações através de junções em geral não são atualizáveis
- Visões definidas usando-se agrupamento e funções de agregação não são atualizáveis

Introdução a Banco de Dados – DCC 011

Restrições de Integridade Gerais

- Comando CREATE ASSERTION

```
CREATE ASSERTION LIMITE_SALARIO  
CHECK (NOT EXISTS(SELECT *  
                     FROM EMPREGADO E, EMPREGADO M,  
                     DEPARTAMENTO D  
                     WHERE E.SALARIO>M.SALARIO AND  
                     E.DNO=D.DNUMERO AND  
                     D.GERSSN=M.SSN));
```

- Cláusula CHECK

```
CREATE DOMAIN D_NUMERO AS INTEGER  
CHECK (D_NUMERO > 0 AND D_NUMERO < 21);
```

Introdução a Banco de Dados – DCC 011

Sumário

- Visão geral da linguagem SQL
 - Breve histórico
 - SQL = LDD + LMD + LCD
 - Comandos de definição de dados
 - CREATE SCHEMA / TABLE / VIEW
 - DROP SCHEMA / TABLE / VIEW
 - ALTER TABLE
 - Comandos de manipulação de dados
 - SELECT, INSERT, UPDATE, DELETE
- Consultas SQL
 - Consultas básicas e avançadas
 - Agrupamento de dados
- Visões em SQL

Introdução a Banco de Dados – DCC 011

Questões e Exercícios Sugeridos

- Capítulo 8:
 - 8.1-8.6, 8.7, 8.9, 8.13
- Capítulo 9:
 - 9.1, 9.2, 9.15

Introdução a Banco de Dados – DCC 011