

# Banco de Dados

Prof. Anthony Ferreira La Marca

[anthony@computacao.cua.ufmt.br](mailto:anthony@computacao.cua.ufmt.br)

# SQL

- Principal motivo para o sucesso dos BD relacionais comerciais
- Linguagem padrão para este tipo de BD
  - Portabilidade entre SGBDs
- Linguagem declarativa
  - Especifica qual deve ser o resultado
  - Deixando otimizações e decisões de como executar a consulta para o SGBD

# SQL

- Inclui alguns recursos de álgebra relacional
- Inclui grande parte dos recursos de cálculo relacional
- Sua sintaxe é muito mais simples que ambas as linguagens formais
- Structured Query Language (Linguagem de Consulta Estruturada)
- Inicialmente chamada de SEQUEL
- Implementada na IBM Research
- Esforço conjunto entre a ANSI e a ISO para alcançarem uma versão padrão
  - Chamada SQL-86

# SQL

- Mais tarde chegaram na SQL-92 (SQL2)
- SQL-1999 (SQL3)
- SQL-2003
- SQL-2006
  - Recursos de XML
- SQL-2008
  - OO
- SQL-2012
  - BI
  - Computação em Nuvem

# SQL

- Fornece instruções de definição de dados (DDL) e consulta e atualizações de dados (DML)
- Define visões
- Segurança e autenticações
- Restrições
- Integridade
- Controle de transações
- Regras para embutir em uma LP

# SQL

- É dividida em 2 especificações
  - Núcleo
    - Deve ser implementado por todos os SGBDR compatíveis com SQL
  - Extensões
    - Implementadas como módulos opcionais
      - Mineração dos dados
      - Dados espaciais
      - Dados temporais
      - Datawarehousing
      - OLAP
      - Dados multimídia

# Termos

- Relação -> tabela
- Tupla -> linha da tabela
- Atributo -> coluna da tabela

# SQL

- Comando CREATE TABLE
- Cria uma relação
- Dando um nome a ela
- Define os atributos, seus tipos e suas restrições iniciais
- Restrições de chave, de integridade de entidade e referencial são especificadas
- Chamadas de tabela de base
- Já a VIEW é chamadas de tabela virtual



# SQL

- Tipos de dados
- Numérico
  - Int
  - Float
  - Double
  - Real
  - Decimal(i,j)
- Cadeia de caracteres
  - Char
  - Varchar
  - CLOB
- Cadeia de BITs
  - BLOB(M) (G)
  - TEXT

# SQL

- Booleano
  - True
  - False
  - UNKNOWN
- Date
  - AAAA-MM-DD
- Time
  - HH-MM-SS
- TimeStamp
  - DD-MM-AAAA -HH-MM-SS

# SQL

- NOT NULL
- PRIMARY KEY
- FOREIGN KEY
- UNIQUE
- DEFAULT
- CHECK
  - Restringir domínio
  - CREATE DOMAIN D\_NUM AS INTEGER  
CHECK (D\_NUM >0 AND D\_NUM <21);

# SQL

- CHECK
  - Restrição baseada em tupla
  - Aplica-se em todas as tuplas individualmente
  - Verificado sempre que uma tupla é inserida ou modificada
  - Aplicado após a definição de um atributo ou domínio
- Exemplo
  - Suponhamos que os números do departamentos sejam inteiro entre 1 e 20

# SQL

```
Dnumero INT NOT NULL CHECK (Dnumero > 0 AND  
Dnumero < 21);
```

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

## CREATE TABLE FUNCIONARIO

(Pnome	VARCHAR(15)	NOT NULL,
Minicial	CHAR,	
Unome	VARCHAR(15)	NOT NULL,
Cpf	CHAR(11),	NOT NULL,
Datanasc	DATE,	
Endereço	VARCHAR(30),	
Sexo	CHAR,	
Salario	DECIMAL(10,2),	
Cpf_supervisor	CHAR(11),	NOT NULL,
Dnr	INT	

PRIMARY KEY (Cpf),

FOREIGN KEY (Cpf\_supervisor) REFERENCES FUNCIONARIO(Cpf),

FOREIGN KEY (Dnr) REFERENCES DEPARTAMENTO(Dnumero) );

## CREATE TABLE DEPARTAMENTO

( Dnome	VARCHAR(15)	NOT NULL,
Dnumero	INT	NOT NULL,
Cpf_gerente	CHAR(11),	NOT NULL,
Data_inicio_gerente	DATE,	

PRIMARY KEY (Dnumero),

UNIQUE (Dnome),

FOREIGN KEY (Cpf\_gerente) REFERENCES FUNCIONARIO(Cpf) );

```

CREATE TABLE LOCALIZACAO_DEP
( Dnumero          INT                      NOT NULL,
  Dlocal            VARCHAR(15)             NOT NULL,
  PRIMARY KEY (Dnumero, Dlocal),
  FOREIGN KEY (Dnumero) REFERENCES DEPARTAMENTO(Dnumero) );

CREATE TABLE PROJETO
( Projnome          VARCHAR(15)             NOT NULL,
  Projnumero        INT                     NOT NULL,
  Projlocal         VARCHAR(15),
  Dnum              INT                     NOT NULL,
  PRIMARY KEY (Projnumero),
  UNIQUE (Projnome),
  FOREIGN KEY (Dnum) REFERENCES DEPARTAMENTO(Dnumero) );

CREATE TABLE TRABALHA_EM
( Fcpf              CHAR(9)                 NOT NULL,
  Pnr               INT                     NOT NULL,
  Horas             DECIMAL(3,1)           NOT NULL,
  PRIMARY KEY (Fcpf, Pnr),
  FOREIGN KEY (Fcpf) REFERENCES FUNCIONARIO(Cpf),
  FOREIGN KEY (Pnr) REFERENCES PROJETO(Projnumero) );

```

**CREATE TABLE DEPENDENTE**

( Fcpf CHAR(11), NOT NULL,

Nome\_dependente VARCHAR(15) NOT NULL,

Sexo CHAR,

Datanasc DATE,

Parentesco VARCHAR(8),

**PRIMARY KEY** (Fcpf, Nome\_dependente),

**FOREIGN KEY** (Fcpf) **REFERENCES** FUNCIONARIO(Cpf) );



# SQL

- Integridade referencial
- Chave estrangeira – Foreign Key
- A restrição de integridade referencial acontece quando tuplas são
  - Inseridas
  - Removidas
  - Alteradas
- Por default o SQL rejeita uma operação que viola a integridade referencial
  - Operação RESTRICT

# SQL

- Há alternativas para o projetista do BD
  - SET NULL
  - CASCADE
  - SET DEFAULT
- Uma opção deve ser qualificada com
  - ON DELETE
  - ON UPDATE
- Por exemplo
  - ON DELETE SET NULL
  - ON UPDATE CASCADE

# SQL

- Se uma tupla para um funcionario supervisor for excluída, o valor CPF\_SUPERVISOR será automaticamente definido como NULL para todas as tuplas que estavam referenciando a tupla do funcionário excluído
- Se o valor de CPF de um funcionário supervisor fosse alterado (por ter sido inserido incorretamente) o valor será propagado para todas as tuplas que referencia ele

# SQL

- Responsabilidade do projetista do BD escolher a ação apropriada
- Em modo geral tabelas de relacionamento e tabela de entidade fracas
- Utilizam CASCADE
- CONSTRAINT – Alias (apelido)

**CREATE TABLE FUNCIONARIO**

```
( ...,  
Dnr          INT          NOT NULL   DEFAULT 1,  
CONSTRAINT CHPFUNC  
    PRIMARY KEY (Cpf),  
CONSTRAINT CHESUPERFUNC  
    FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf)  
    ON DELETE SET NULL   ON UPDATE CASCADE,  
CONSTRAINT CHEDEPFUNC  
    FOREIGN KEY(Dnr) REFERENCES DEPARTAMENTO(Dnumero)  
    ON DELETE SET DEFAULT ON UPDATE CASCADE);
```

**CREATE TABLE DEPARTAMENTO**

```
( ...,  
Cpf_gerente  CHAR(11)     NOT NULL   DEFAULT '88866555576',  
...,  
CONSTRAINT CHPDEP  
    PRIMARY KEY(Dnumero),  
CONSTRAINT CHSDEP  
    UNIQUE (Dnome),  
CONSTRAINT CHEGERDEP  
    FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf)  
    ON DELETE SET DEFAULT ON UPDATE CASCADE);
```

**CREATE TABLE LOCALIZACAO\_DEP**

```
( ...,  
PRIMARY KEY (Dnumero, Dlocal),  
FOREIGN KEY (Dnumero) REFERENCES DEPARTAMENTO(Dnumero)  
    ON DELETE CASCADE   ON UPDATE CASCADE);
```

# SQL

- Operador DROP
- Remove elementos nomeados do esquema
  - Tabelas
  - Domínios
  - Restrições
- Exemplo
  - Remover a tabela DEPENDENTE
  - DROP TABLE DEPENDENTE CASCADE;

# SQL

- Com a opção CASCADE
  - Todas as restrições, views e elementos que referenciam a tabela sendo removida também serão excluídos automaticamente do esquema
- Com a opção RESTRICT
  - Só será excluída caso a tabela não for referenciada em quaisquer restrições
  - Por exemplo
    - Chave estrangeira com outra relação

# SQL

- Além de remover a tabela ele exclui a definição da tabela do catálogo do SGBD
- Assim não pode-se utiliza-la futuramente
- Caso queira apenas remover os registros
- Utilize o comando DELETE
- DROP usado também para remover domínios e restrições



# SQL

- Operadores que alteram um esquema
- Acrescentam ou removem
  - Tabelas
  - Atributos
  - Restrições
- Alterações estas que podem ser feitas enquanto o BD esta operando
- SGBD verifica se tais alterações não acarretará em inconsistência dos dados

# SQL

- Alteração feita através do operador ALTER
- ALTER TABLE
- Acrescentar ou remover uma coluna (atributo)
- Alterar uma definição de coluna
- Acrescentar ou remover restrições de tabelas
- Exemplo

# SQL

- Incluir um atributo que mantém as tarefas dos funcionários na relação da base FUNCIONARIO
- ALTER TABLE FUNCIONARIO  
    ADD COLUMN TAREFA VARCHAR(12);
- Insere valor NULL por padrão
- Logo a restrição NOT NULL não é permitida
- Na remoção deve escolher entre CASCADE ou RESTRICT

# SQL

- Caso CASCADE
  - Todas as restrições e views que referenciam a coluna são removidas
- Caso RESTRICT
  - Apenas excluí caso a coluna não seja referenciada por nenhuma view ou relação
- Exemplo
  - Remover o atributo tarefa que foi inserido
  - ALTER TABLE FUNCIONARIO  
DROP COLUMN TAREFA CASCADE;

# SQL

- Na alteração da definição de um atributo
- Exemplo
  - O campo endereço é varchar(50) ao invés de 30
  - ALTER TABLE FUNCIONARIO  
MODIFY ENDERECO VARCHAR(50);
- Para mudar o nome da tabela e do atributo respectivamente
  - ALTER TABLE FUNCIONARIO RENAME TO FUNC;
  - ALTER TABLE FUNC CHANGE COLUMN ENDERECO  
END VARCHAR(50);

# SQL

- Para adicionar e remover uma restrição de não nulo a uma coluna, respectivamente
  - ALTER TABLE FUNC CHANGE MINICIAL INICIAL CHAR NOT NULL;
  - ALTER TABLE FUNCIONARIO CHANGE INICIAL MINICIAL CHAR;
- Para adicionar uma restrição de chave estrangeira a uma tabela
  - ALTER TABLE TRABALHA\_EM ADD CONSTRAINT FK\_CPC FOREIGN KEY (F\_CPF) REFERENCES FUNCIONARIO (CPF);

# SQL

- Para adicionar uma restrição de chave primaria
  - ALTER TABLE FUNCIONARIO ADD PRIMARY KEY (CPF);
- Para remover uma restrição, esta precisa ser atribuída um nome (CONSTRAINT)
  - ALTER TABLE FUNC DROP FOREIGN KEY CPF\_FK CASCADE;
  - ALTER TABLE FUNC DROP PRIMARY KEY; APELIDO DA FK

# Exercício

- Considere a seguinte Figura
- Escrevas as instruções DDL da SQL para declarar o esquema do banco de dados relacional UNIVERSIDADE. Especifique as chaves e as ações (comportamentos) de disparo referencial
- Utilize o ALTER para definir algumas FK

## ALUNO

Nome	Numero_aluno	Tipo_aluno	Curso
Silva	17	1	CC
Braga	8	2	CC

## DISCIPLINA

Nome_disciplina	Numero_disciplina	Creditos	Departamento
Introd. à ciência da computação	CC1310	4	CC
Estruturas de dados	CC3320	4	CC
Matemática discreta	MAT2410	3	MAT
Banco de dados	CC3380	3	CC

## TURMA

Identificacao_turma	Numero_disciplina	Semestre	Ano	Professor
85	MAT2410	Segundo	07	Kleber
92	CC1310	Segundo	07	Anderson
102	CC3320	Primeiro	08	Carlos
112	MAT2410	Segundo	08	Chang
119	CC1310	Segundo	08	Anderson
135	CC3380	Segundo	08	Santos

## HISTORICO\_ESCOLAR

Numero_aluno	Identificacao_turma	Nota
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

## PRE\_REQUISITO

Numero_disciplina	Numero_pre_requisito
CC3380	CC3320
CC3380	MAT2410
CC3320	CC1310



# Exercício

- 2 – Considere o esquema abaixo de uma biblioteca e utilize comandos DDL para criá-los. Defina os comportamentos para a integridade referencial



# SQL

- SELECT
  - Recupera informação do BD
  - Retorna um multiconjunto de tuplas (bag)
- SELECT básico
  - SELECT <lista de atributos>
  - FROM < lista de tabelas>
  - WHERE <condição>;
- Onde

# SQL

- Lista de atributos
  - Os atributos a serem recuperados pela consulta
- Lista de tabelas
  - Os nomes das relações exigidos para processar a consulta
- Condição
  - Uma expressão condicional que identifica as tuplas a serem recuperadas pela consulta

# SQL

- Pode existir nomes de atributos iguais em relações diferentes
- Porém se ambas estiverem na mesmas consulta
- Deve-se colocar o nome da relação ponto atributo
  - Relação.atributo
- Desta forma é possível identificar qual é qual

# SQL

- Há também a possibilidade de criar-se alias (apelidos) para as tabelas
- Evitando assim, a necessidade de se repetir nomes longos de tabelas
- Exemplo:
  - Mostrar o nome, sobrenome e o endereço dos funcionários que trabalham no departamento de Pesquisa

```
SELECT F.PNOME, F.UNOME, F.ENDERECO  
FROM FUNCIONARIO F, DEPARTAMENTO D  
WHERE D.DNOME LIKE "%PESQUISA%" AND  
F.DNR = D.DNUMERO
```

# SQL

- Quando uma consulta se refere a uma mesma relação duas vezes, use-se alias
- Exemplo:
  - Para cada funcionário, recupere o primeiro nome e o último nome e idem ao seu supervisor imediato

```
SELECT F.PNOME, F.UNOME, S.PNOME, S.UNOME  
FROM FUNCIONARIO F, FUNCIONARIO S  
WHERE F.CPF_SUPERVISOR = S.CPF;
```

- Chamada consulta recursiva

# SQL

- \*
  - Recupera todos os atributos que estão nas relações depois da cláusula FROM
  - `SELECT * FROM FUNCIONARIO WHERE DNR =5;`
- DISTINCT
  - Elimina tuplas repetidas de acordo com o atributo que foi usado a cláusula DISTINCT
  - Recuperar os salários distintos dos funcionários
  - `SELECT DISTINCT SALARIO FROM FUNCIONARIO;`
- Há também as operações UNION (união), EXCEPT (diferença) e INTERSECT (interseção)

# SQL

- Comparações de cadeia de caracteres
  - Operador LIKE
- Pegar cadeias parciais
  - Operador %
- Exemplo
  - Recuperar todos os funcionários cujo endereço esteja em Barra do Garças, MT
  - `SELECT F.PNOME, F.UNOME FROM FUNCIONARIO F WHERE F.ENDERECO LIKE '%Barra do Garças,MT%';`



# SQL

- Agora precisa-se recuperar todos os funcionários que nasceram no ano de 1965
- Utilização do operador LIKE (comparar) +
- Operador \_
  - Substitui um único caracter
- ```
SELECT F.PNOME, F.UNOME  
FROM FUNCIONARIO F  
WHERE F.DATANASC LIKE '__ 6 5- __ -__';
```
- Se houver a necessidade de utilizar % e \_ como caractere literal (normal)
  - Operador ESCAPE

# SQL

- Operador CONCAT
  - Concatenas strings
- Operador BETWEEN
  - Pega o intervalo
- Exemplo
  - Mostrar o nome e o sobrenome dos funcionários em um único campo que ganham entre R\$1000,00 e R\$2000,00
  - `SELECT CONCAT (F.PNOME, ' ', F.UNOME)`  
`FROM FUNCIONARIO F`  
`WHERE (F.SALARIO BETWEEN 1000 AND 2000);`

# SQL

- Operador AS
- Exemplo
  - Mostrar o salário resultante de cada funcionário que trabalha no projeto 'Universidade' receber um aumento de 10%
  - `SELECT F.PNOME, F.UNOME, F.SALARIO *1.1 AS  
AUMENTA SALARIO  
  
FROM FUNCIONARIO F, TRABALHA_EM T, PROJETO P  
WHERE F.CPF = T.FCPF AND T.PNR = P.PROJNUMERO  
AND P.PROJNOME LIKE 'UNIVERSIDADE';`

# SQL

- Operadores
- Upper()
  - Passa toda a string para maiúsculo
- Lower()
  - Passa toda a string para minúsculo
- Distinct
  - Elimina duplas duplicadas da consulta

# SQL

- Operador para ordenar as tuplas do resultado de uma consulta
- Ordenar pelos valores de um ou mais atributos
- ORDER BY
- Exemplo
  - Recuperar uma lista de funcionários e dos projetos em que estão trabalhando, ordenada por departamento e, dentro de cada departamento, ordenada pelo sobrenome e depois pelo nome

# SQL

- SELECT D.DNOME, F.UNOME, F.PNOME,  
P.PROJNOME  
FROM DEPARTAMENTO D, FUNCIONARIO F,  
PROJETO P, TRABALHA\_EM T  
WHERE D.DNUMERO = F.DNR AND F.CPF =  
T.FCPF AND T.PNR = P.PROJNUMERO  
ORDER BY D.DNOME, F.UNOME, F.PNOME;
- Por padrão ordena de forma crescente
- Para ordenar de forma decrescente
  - Operador DESC depois do atributo

# SQL

- Realiza algumas operações de conjunto
- União de conjuntos
  - Union
- Diferença de conjuntos
  - Except ou Minus (depende do SGBD)
- Intersecção de conjuntos
  - intersect
- Consultas mais eficientes e eliminam tuplas duplicadas

# SQL

- A operação UNION requer que as relações envolvidas tenham a mesma quantidade de atributos e seus respectivos pertencentes ao mesmo domínio
- Exemplo
  - Fazer uma lista de todos os números de projetos para aqueles que envolvam um funcionário cujo o último nome é “Silva”, seja um trabalhador ou como um gerente do departamento que controla o projeto



# SQL

- SELECT DISTINCT P.PROJNUMERO  
FROM PROJETO P, DEPARTAMENTO D,  
FUNCIONARIO F,  
WHERE P.DNUM = D.DNUMERO AND  
D.CPF\_GERENTE = F.CPF AND F.UNOME = "SILVA";
- UNION
- SELECT DISTINCT P.PROJNUMERO  
FROM PROJETO P, TRABALHA\_EM T,  
FUNCIONARIO F  
WHERE P.PROJNUMERO = T.PNR AND F.CPF =  
T.FCPF AND F.UNOME = "SILVA";

# SQL

|     |    |    |     |    |     |    |
|-----|----|----|-----|----|-----|----|
| (a) | R  | S  | (b) | T  | (c) | T  |
|     | A  | A  |     | A  |     | A  |
|     | a1 | a1 |     | a1 |     | a2 |
|     | a2 | a2 |     | a1 |     | a3 |
|     | a2 | a4 |     | a2 |     |    |
|     | a3 | a5 |     | a2 |     |    |
|     |    |    |     | a2 |     |    |
|     |    |    |     | a3 |     |    |
|     |    |    |     | a4 |     |    |
|     |    |    |     | a5 |     |    |
|     |    |    |     |    | (d) | T  |
|     |    |    |     |    |     | A  |
|     |    |    |     |    |     | a1 |
|     |    |    |     |    |     | a2 |

Figura 4.5

Os resultados das operações de multiconjunto da SQL. (a) Duas tabelas, R(A) e S(A). (b) R(A) UNION ALL S(A). (c) R(A) EXCEPT ALL S(A). (d) R(A) INTERSECT ALL S(A).

# Exercícios

- Especifique as consultas em SQL
- Recupere o nome de todos os alunos que estão cursando 'CC'
- Recupere os nome de todas as disciplinas lecionadas pelo professor 'kleber' em 2007 e 2008
- Para cada matéria lecionada pelo professor 'kleber', recupere o número da disciplina, semestre, ano e o número das disciplinas que são pré requisitos

## ALUNO

| Nome  | Numero_aluno | Tipo_aluno | Curso |
|-------|--------------|------------|-------|
| Silva | 17           | 1          | CC    |
| Braga | 8            | 2          | CC    |

## DISCIPLINA

| Nome_disciplina                 | Numero_disciplina | Creditos | Departamento |
|---------------------------------|-------------------|----------|--------------|
| Introd. à ciência da computação | CC1310            | 4        | CC           |
| Estruturas de dados             | CC3320            | 4        | CC           |
| Matemática discreta             | MAT2410           | 3        | MAT          |
| Banco de dados                  | CC3380            | 3        | CC           |

## TURMA

| Identificacao_turma | Numero_disciplina | Semestre | Ano | Professor |
|---------------------|-------------------|----------|-----|-----------|
| 85                  | MAT2410           | Segundo  | 07  | Kleber    |
| 92                  | CC1310            | Segundo  | 07  | Anderson  |
| 102                 | CC3320            | Primeiro | 08  | Carlos    |
| 112                 | MAT2410           | Segundo  | 08  | Chang     |
| 119                 | CC1310            | Segundo  | 08  | Anderson  |
| 135                 | CC3380            | Segundo  | 08  | Santos    |

## HISTORICO\_ESCOLAR

| Numero_aluno | Identificacao_turma | Nota |
|--------------|---------------------|------|
| 17           | 112                 | B    |
| 17           | 119                 | C    |
| 8            | 85                  | A    |
| 8            | 92                  | A    |
| 8            | 102                 | B    |
| 8            | 135                 | A    |

## PRE\_REQUISITO

| Numero_disciplina | Numero_pre_requisito |
|-------------------|----------------------|
| CC3380            | CC3320               |
| CC3380            | MAT2410              |
| CC3320            | CC1310               |

# Exercício

- Recupere o nome e histórico de cada aluno sênior (Tipo\_aluno = 2) formando em CC. Um histórico inclui nome da disciplina, número da disciplina, crédito, semestre, ano e nota para cada disciplina concluída pelo aluno

# SQL

- Três comandos utilizados para modificar o BD
- INSERT
  - Acrescentar uma nova tupla à relação
- UPDATE
  - Alterar uma tupla de uma relação
- DELETE
  - Deletar uma tupla de uma relação

# SQL

- INSERT
- Especificar o nome da relação
- E os valores na mesma ordem dos atributos
- Exemplo
  - Acrescentar um novo funcionário na relação FUNCIONARIO
  - INSERT INTO FUNCIONARIO VALUES ('32145678989', 'Ricardo', 'S', 'Borges', '1962-12-22', 'Rua Itapevi, 44, Cuiaba,MT', M, 5600, '32145678989', 5);

# SQL

- Uma segunda forma é especificar também os atributos
- Exemplo
  - INSERT INTO FUNCIONARIO (CPF, PNOME, DNR, CPF\_SUPERVISOR) VALUES ('32145678989', 'Ricardo', 5, '32145678989');

# SQL

- Se o SGBD garante a integridade referencial as seguintes inserções serão ignoradas?
- `INSERT INTO FUNCIONARIO (CPF, PNOME, DNR, CPF_SUPERVISOR) VALUES ('32145678989', 'Ricardo', 2, '32145678989');`
- `INSERT INTO FUNCIONARIO (PNOME, DNR) VALUES ('Ricardo', 5);`



# SQL

- Pode-se utilizar um SELECT dentro de um INSERT
- Assim será incluída todas as tuplas que satisfazem a condição
- Exemplo
  - Criar uma tabela temporária para armazenar o nome do funcionário, o nome do projeto e as horas por semana para cada funcionário que trabalha no projeto
  - CREATE TABLE TRABALHA\_EM\_INFO (  
    FUNC\_NOME                      VARCHAR (12),  
    PROJ\_NOME                     VARCHAR (12),  
    HORA\_SEMANAL                 DECIMAL(3,1));

# SQL

- INSERT INTO TRABALHA\_EM\_INFO  
(FUNC\_NOME, PROJ\_NOME,  
HORA\_SEMANAL)

```
        SELECT F.PNOME, P.PROJNOME, T.HORAS  
        FROM FUNCIONARIO F, PROJETO P,  
TRABALHA_EM T  
        WHERE P.PROJNUMERO = T.PNR AND  
T.FCPF = F.CPF;
```

# SQL

- OBS: A tabela TRABALHA\_EM\_INFO NÃO SERÁ ATUALIZADA caso as tabelas envolvidas sejam!
- Para que ela seja, utilize VIEW
- Veremos mais para frente

# SQL

- DELETE
- Utiliza a cláusula WHERE
  - Podendo se estender a outras relações dependendo do COMPORTAMENTO DO DELETE escolhido (CASCADE, RESTRICT, SET NULL, SET DEFAULT)
- Se não haver a cláusula WHERE, todas as tuplas da relação serão excluídas
- Entretanto, a tabela continua no BD, porém vazia

# SQL

- Exemplo
  - DELETE FROM FUNCIONARIO  
WHERE PNOME = 'Ricardo';
  - DELETE FROM FUNCIONARIO  
WHERE DNR = 5;
  - DELETE FROM FUNCIONARIO;

# SQL

- UPDATE
- Utilizado para modificar o valor de atributos de uma ou mais tuplas
- A cláusula WHERE seleciona as tuplas a serem modificadas
- Mudança na chave primária pode ser propagada para os valores das chaves estrangeiras se utilizar o COMPORTAMENTO UPDATE correto
- O comando SET especifica os atributos a serem modificados e os seus novos valores

# SQL

- Exemplo
- Alterar o local e o número de departamento que controla o projeto número 10 para 'Santo André' e 5
- UPDATE PROJETO  
    SET PROJLOCAL = 'Santo André', DNUM = 5  
    WHERE PROJNUM = 10;

# SQL

- Várias tuplas podem ser modificadas a partir de um único comando UPDATE
- Exemplo
  - Dar 10% de aumento para todos os funcionários que trabalham no departamento de 'Pesquisa'
  - UPDATE FUNCIONARIO F SET SALARIO = SALARIO \* 1.1  
WHERE F.DNR = (SELECT DISTINCT D.DNUMERO  
FROM DEPARTAMENTO D  
WHERE D.DNOME LIKE 'Pesquisa');



# Exercício

- Faça as inserções em todas as tabelas do esquema UNIVERSIDADE
- Cuidado com as chaves primárias e estrangeiras
- Escreve instruções de atualizações SQL para realizar ações sobre as instâncias do BD mostrado ao lado

## ALUNO

| Nome  | Numero_aluno | Tipo_aluno | Curso |
|-------|--------------|------------|-------|
| Silva | 17           | 1          | CC    |
| Braga | 8            | 2          | CC    |

## DISCIPLINA

| Nome_disciplina                 | Numero_disciplina | Creditos | Departamento |
|---------------------------------|-------------------|----------|--------------|
| Introd. à ciência da computação | CC1310            | 4        | CC           |
| Estruturas de dados             | CC3320            | 4        | CC           |
| Matemática discreta             | MAT2410           | 3        | MAT          |
| Banco de dados                  | CC3380            | 3        | CC           |

## TURMA

| Identificacao_turma | Numero_disciplina | Semestre | Ano | Professor |
|---------------------|-------------------|----------|-----|-----------|
| 85                  | MAT2410           | Segundo  | 07  | Kleber    |
| 92                  | CC1310            | Segundo  | 07  | Anderson  |
| 102                 | CC3320            | Primeiro | 08  | Carlos    |
| 112                 | MAT2410           | Segundo  | 08  | Chang     |
| 119                 | CC1310            | Segundo  | 08  | Anderson  |
| 135                 | CC3380            | Segundo  | 08  | Santos    |

## HISTORICO\_ESCOLAR

| Numero_aluno | Identificacao_turma | Nota |
|--------------|---------------------|------|
| 17           | 112                 | B    |
| 17           | 119                 | C    |
| 8            | 85                  | A    |
| 8            | 92                  | A    |
| 8            | 102                 | B    |
| 8            | 135                 | A    |

## PRE\_REQUISITO

| Numero_disciplina | Numero_pre_requisito |
|-------------------|----------------------|
| CC3380            | CC3320               |
| CC3380            | MAT2410              |
| CC3320            | CC1310               |

# Exercício

- 1 – Inserir um novo aluno ‘Alvez’, 25, 1, ‘MAT’, no BD
- 2 – Alterar a nota para ‘A’ do aluno ‘Silva’ da turma 112
- 3 – Inserir uma nova disciplina ‘Engenharia do Conhecimento’, ‘CC4390’, 3, ‘CC’
- 5 – Excluir o registro para o aluno cujo nome é ‘Silva’ e cujo o número de aluno é 17.
- 4 – Alterar o professor da turma do aluno ‘Braga’ para “Ivairton” cuja a disciplina é ‘ED’