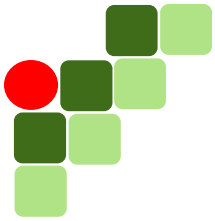
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Araraquara

Banco de Dados II

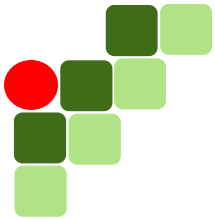
Cristiane Yaguinuma
cristiane.yaguinuma@ifsp.edu.br

- Revisão SQL – DDL e DML



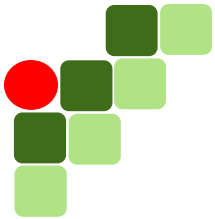
Roteiro da aula

- Revisão SQL
  - Comandos DDL
    - Tipos de dados básicos
    - Restrições
  - Comandos DML



Comandos DDL

- DDL: Data Definition Language
- Permite ao usuário definir a estrutura e organização dos dados armazenados, além das relações que existem entre eles
- Atuam sobre os objetos de bancos de dados
- Os objetos podem ser criados (CREATE), alterados (ALTER) e removidos (DROP)

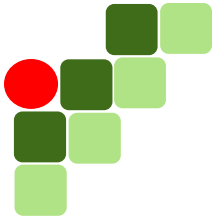


CREATE TABLE

- CREATE TABLE
  - Criar tabelas em um banco de dados
  - Usuário deve ter privilégio CREATE TABLE

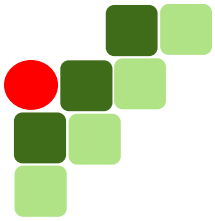
Sintaxe básica:

CREATE TABLE *nome_tabela*
(<Definições de colunas>, …
 <Restrições> …
);



Tipos de dados básicos

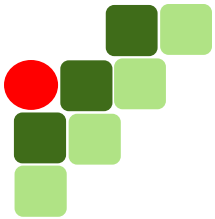
Data Type	Description
VARCHAR2(*size*)	Variable-length character data
CHAR(*size*)	Fixed-length character data
NUMBER(*p*,*s*)	Variable-length numeric data
DATE	Date and time values
LONG	Variable-length character data (up to 2 GB)
CLOB	Character data (up to 4 GB)
RAW and LONG RAW	Raw binary data
BLOB	Binary data (up to 4 GB)



DEFAULT

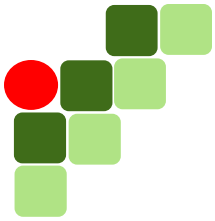
- Especifica valores padrão para uma coluna
  - Valores literais
  - Expressões
  - Funções SQL
- O valor deve corresponder ao tipo de dado da coluna

```
CREATE TABLE DEPT
( dnumber       NUMBER(6) PRIMARY KEY NOT NULL,
  dname         VARCHAR2(30) NOT NULL,
  creation_date      DATE DEFAULT SYSDATE
);
```



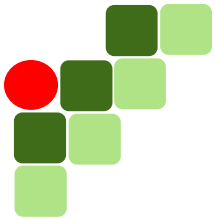
Restrições de integridade

- NOT NULL
- PRIMARY KEY (<Atributo>,...)
- UNIQUE (<Atributo>,...)
- FOREIGN KEY (<Atributo>,...) REFERENCES <tabela> (<Atributos chaves>, ...)
- CHECK



CREATE TABLE – Restrições

```
CREATE TABLE EMPLOYEES
( employee_id          NUMBER(6)        PRIMARY KEY,
  last_name            VARCHAR2(25)     NOT NULL,
  email                VARCHAR2(25),
  hire_date            DATE             NOT NULL,
  salary               NUMBER(8,2),
  commission_pct       NUMBER(8,2),
  department_id        NUMBER(4),
  [CONSTRAINT emp_dept_fk]
     FOREIGN KEY (department_id)
     REFERENCES DEPARTMENTS(department_id),
  [CONSTRAINT emp_email] UNIQUE(email),
  [CONSTRAINT emp_salary] CHECK (salary > 0)
);
```



Restrição de integridade referencial

- Integridade referencial → FOREIGN KEY
  - Coluna(s) de uma tabela referenciam coluna(s) de outra(s) tabela(s)

- Pode ser violada quando:
  - Tuplas são inseridas ou excluídas
  - Atualização de um valor de chave primária ou chave estrangeira

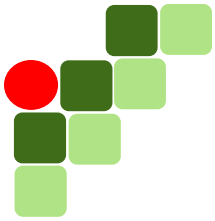
- Ação padrão do SGBD é rejeitar a operação que viola a integridade referencial

Restrição de integridade referencial

- Ação de disparo referencial
  - Ação alternativa para ser realizada quando uma restrição de chave estrangeira é violada

- ON DELETE
  - CASCADE
  - SET NULL

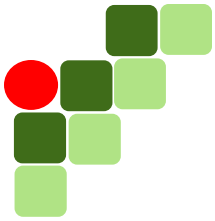
- ON UPDATE
  - CASCADE



Restrições com FOREIGN KEY

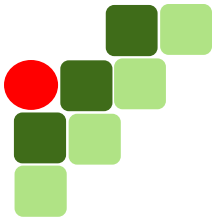
- ON DELETE CASCADE
  - Se uma tupla é apagada, todas as tuplas que a referenciam também são apagadas

- ON DELETE SET NULL
  - Se uma tupla é apagada, todas as tuplas que a referenciam recebem valor nulo na FK



Restrições com FOREIGN KEY

```
CREATE TABLE emp_test
(employee_id NUMBER(6)  PRIMARY KEY,
 name VARCHAR2(50)      NOT NULL,
 manager_id NUMBER(4)
      CONSTRAINT fk_mgr
      REFERENCES employees ON DELETE SET NULL,
 salary NUMBER(8,2),
 department_id NUMBER(2)
      CONSTRAINT fk_deptno
      REFERENCES departments(department_id)
      ON DELETE CASCADE
);
```



Restrições com FOREIGN KEY

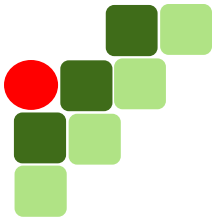
- ON UPDATE CASCADE
  - Se uma tupla tem valor de PK atualizado, esta atualização é propagada para todas as tuplas que a referenciam

- Não é oferecida no Oracle
  - Alternativa: usar gatilho (trigger)

CREATE TABLE usando subconsulta

```
CREATE TABLE dept_copy
AS
     SELECT *
     FROM DEPARTMENTS;
```

```
CREATE TABLE emp_copy
AS
     SELECT *
     FROM EMPLOYEES;
```

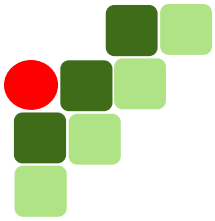


CREATE TABLE usando subconsulta

```
CREATE TABLE dept_80
      AS
         SELECT employee_id, last_name,
                salary*12 annual_salary,
                hire_date
         FROM EMPLOYEES
         WHERE department_id = 80;
```

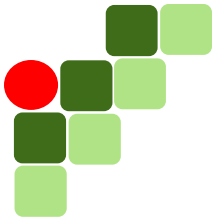
```
DESCRIBE dept_80;

SELECT * FROM dept_80;
```



Revisão: comandos DML

- DML: Data Manipulation Language
- Permitem manipular o estado do banco de dados
  - INSERT INTO
  - UPDATE
  - DELETE

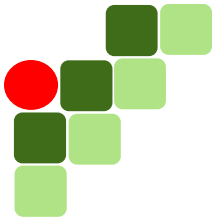


INSERT INTO

- Adiciona novos dados às tabelas

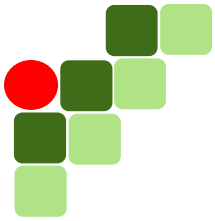
- Sintaxe básica:
- INSERT INTO <Tabela> [(<Atr1>, <Atr2>, …)]
  VALUES (<Valor_Atr1>, <Valor_Atr2>, …);

```
INSERT INTO dept_copy(department_id, department_name,
manager_id, location_id)
VALUES (998, 'Press', 100, 1700);
```



INSERT INTO usando subconsulta

```
INSERT INTO dept_80(employee_id, last_name, annual_sal,
                    hire_date)
   SELECT employee_id, last_name, salary*12, hire_date
   FROM EMPLOYEES
   WHERE department_id > 80;
```



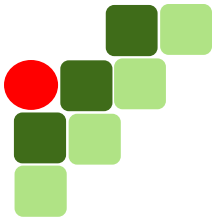
UPDATE

- Efetua mudança (atualização) nos dados

- Sintaxe:

  UPDATE <tabela>
  SET <Atributo> = <expressão>, ...
  [WHERE <Condição>]

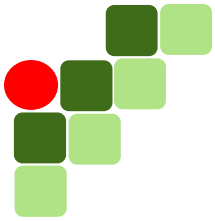
```
UPDATE emp_copy
SET commission_pct = NULL
WHERE job_id = 'SH_CLERK';
```



UPDATE

- É possível atualizar valores com base em valores de outras tabelas

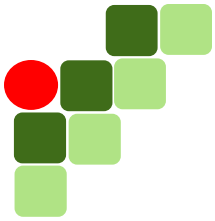
```
UPDATE emp_copy
SET job_id = (SELECT job_id
              FROM employees
              WHERE employee_id = 205),
    department_id = (SELECT department_id
                     FROM DEPARTMENTS
                     WHERE department_name = 'Sales')
WHERE employee_id > 200;
```



UPDATE

- É possível atualizar valores com base em valores de outras tabelas

```
UPDATE emp_copy
SET salary = (SELECT salary
              FROM employees
              WHERE employee_id = 110),
WHERE department_id = (SELECT department_id
                       FROM DEPARTMENTS
                       WHERE department_name = 'Sales');
```



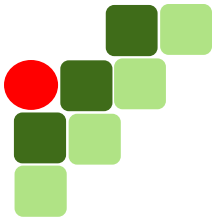
DELETE

- Remove uma ou mais linhas de uma tabela no banco de dados

- Sintaxe:

  DELETE [FROM] <tabela>
  [WHERE <Condição>]

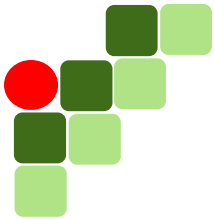
```
DELETE FROM emp_copy
WHERE salary < 3000;
```



DELETE

```
DELETE FROM dept_80;
```

```
DELETE FROM emp_copy
WHERE department_id =
      (SELECT department_id
       FROM departments
       WHERE department_name LIKE 'Sales');
```



TRUNCATE

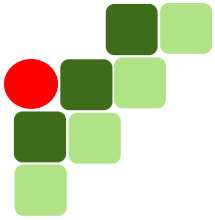
- É um comando DDL que remove todas as linhas de uma tabela

- É mais eficiente que DELETE FROM
  - Não dispara triggers
  - Não guarda dados para rollback
    - Portanto, é difícil de desfazer

```
TRUNCATE TABLE emp_copy;
```

```
TRUNCATE TABLE dept_copy;
```

Dúvidas?



Exercícios

- Lista de exercícios 1 – Revisão SQL

Referências

- **Documentação Oracle**
  - SQL Reference

- ELMASRI, Ramez E.; NAVATHE, Shamkant B. **Sistemas de Banco de Dados.** 6.ed. Pearson, 2011.