Title: Working With Constraints

Author: S. Santhosh Kumar

Objectives

After the completion of this exercise, the students will be able to:

- Describe the constraints
- Create and maintain the constraints

What are Integrity Constraints?

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table if there are dependencies.

Types of Integrity Constraints

a) Domain Integrity

- NOT NULL
- CHECK

b) Entity Integrity

- UNIQUE
- PRIMARY KEY

c) Referential Integrity

- FOREIGN KEY

Creating Constraints

Constraints can be created in two ways:

1. At the same time as the table is created.
2. After the table has been created.

Syntax:

```
CREATE TABLE tablename (
  column_name1 data_type constraints,
  column_name2 data_type constraints,
  ...
);
```

Example:

```
CREATE TABLE employees (
  employee_id number(6),
  first_name varchar2(20),
  job_id varchar2(10),
  CONSTRAINT emp_emp_id_pk PRIMARY KEY (employee_id)
);
```

Output:

```
Table created.
```

Working with Constraints

Domain Integrity

This constraint sets a range and any violations will prevent the manipulation that caused the breach.

NOT NULL Constraint

```
CREATE TABLE employees (
  employee_id number(6),
  last_name varchar2(25) NOT NULL,
  salary number(8,2),
  commission_pct number(2,2),
  hire_date date CONSTRAINT emp_hire_date_nn NOT NULL
);
```

Output:

```
Table created.
```

CHECK Constraint

```
CREATE TABLE employees (
  employee_id number(6),
  last_name varchar2(25) NOT NULL,
  salary number(8,2),
  commission_pct number(2,2),
  hire_date date CONSTRAINT emp_hire_date_nn NOT NULL,
  CONSTRAINT emp_salary_mi CHECK(salary > 0)
);
```

Output:

```
Table created.
```

Entity Integrity

Unique Key Constraint

```
CREATE TABLE employees (
  employee_id number(6),
  last_name varchar2(25) NOT NULL,
  email varchar2(25),
  salary number(8,2),
  commission_pct number(2,2),
  hire_date date CONSTRAINT emp_hire_date_nn NOT NULL,
  CONSTRAINT emp_email_uk UNIQUE(email)
);
```

Output:

```
Table created.
```

Primary Key Constraint

```
CREATE TABLE employees (
  employee_id number(6),
  last_name varchar2(25) NOT NULL,
  email varchar2(25),
  salary number(8,2),
  commission_pct number(2,2),
  hire_date date CONSTRAINT emp_hire_date_nn NOT NULL,
  CONSTRAINT emp_id_pk PRIMARY KEY (employee_id),
  CONSTRAINT emp_email_uk UNIQUE(email)
);
```

Output:

```
Table created.
```

Referential Integrity

Foreign Key

```
CREATE TABLE employees (

  employee_id number(6),
  last_name varchar2(25) NOT NULL,
  email varchar2(25),
  salary number(8,2),
  commission_pct number(2,2),
  hire_date date CONSTRAINT emp_hire_date_nn NOT NULL,
  CONSTRAINT emp_id_pk PRIMARY KEY (employee_id),
  CONSTRAINT emp_email_uk UNIQUE(email),
  CONSTRAINT emp_dept_fk FOREIGN KEY (department_id) REFERENCES
departments(dept_id)
);
```

Output:

```
Table created.
```

Adding and Dropping Constraints

Adding a Constraint

```
ALTER TABLE employees ADD CONSTRAINT emp_manager_fk FOREIGN KEY
(manager_id) REFERENCES employees (employee_id);
```

Output:

```
Table altered.
```

Dropping a Constraint

```
ALTER TABLE employees DROP CONSTRAINT emp_manager_fk;
```

Output:

```
Table altered.
```

Cascade in Drop

```
ALTER TABLE departments DROP PRIMARY KEY CASCADE;
```

Output:

```
Table altered.
```

Enabling and Disabling Constraints

Disabling Constraints

```
ALTER TABLE employees DISABLE CONSTRAINT emp_emp_id_pk CASCADE;
```

Output:

```
Table altered.
```

Enabling Constraints

```
ALTER TABLE employees ENABLE CONSTRAINT emp_emp_id_pk CASCADE;
```

Output:

```
Table altered.
```

Viewing Constraints

Query the USER_CONSTRAINTS Table

```
SELECT constraint_name, constraint_type, search_condition
FROM user_constraints
WHERE table_name = 'employees';
```

Output:

```
CONSTRAINT_NAME   CONSTRAINT_TYPE   SEARCH_CONDITION
----------------  ----------------  ---------------------------------------
EMP_EMP_ID_PK     P
EMP_EMAIL_UK      U
EMP_DEPT_FK       R                 "DEPARTMENT_ID" IS NOT NULL
```

Query the USER_CONS_COLUMNS Table

```
SELECT constraint_name, constraint_type
FROM user_cons_columns
WHERE table_name = 'employees';
```

Output:

```
CONSTRAINT_NAME   CONSTRAINT_TYPE
----------------  ---------------
EMP_EMP_ID_PK     P
EMP_EMAIL_UK      U
EMP_DEPT_FK       R
```

Exercises

1. **Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. Name the constraint my_emp_id_pk.**

```
ALTER TABLE emp ADD CONSTRAINT my_emp_id_pk PRIMARY KEY (id);
```

Output:

```
Table altered.
```

2. **Create a PRIMARY KEY constraint on the DEPT table using the ID column. Name the constraint my_dept_id_pk.**

```
ALTER TABLE dept ADD CONSTRAINT my_dept_id_pk PRIMARY KEY (id);
```

Output:

```
Table altered.
```

3. **Add a column DEPT_ID to the EMP table. Add a foreign key reference ensuring the employee is not assigned to a nonexistent department. Name the constraint my_emp_dept_id_fk.**

```
ALTER TABLE emp ADD dept_id NUMBER;
ALTER TABLE emp ADD CONSTRAINT my_emp_dept_id_fk FOREIGN KEY
(dept_id) REFERENCES dept (id);
```

Output:

```
Table altered.
```

4. **Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to ensure the commission value is greater than zero.**

```
ALTER TABLE emp ADD commission NUMBER(2,2);
ALTER TABLE emp ADD CONSTRAINT emp_commission_ck CHECK (commission >
0);
```

Output:

```
Table altered.
```