

# 10

## Including Constraints

# Objectives

**After completing this lesson, you should be able to do the following:**

- **Describe constraints**
- **Create and maintain constraints**

# What are Constraints?

- **Constraints enforce rules at the table level.**
- **Constraints prevent the deletion of a table if there are dependencies.**
- **The following constraint types are valid:**
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

# Constraint Guidelines

- Name a constraint or the Oracle server generates a name by using the `SYS_Cn` format.
- Create a constraint either:
  - At the same time as the table is created, or
  - After the table has been created
- Define a constraint at the column or table level.
- View a constraint in the data dictionary.

# Defining Constraints

```
CREATE TABLE [schema.]table
    (column datatype [DEFAULT expr]
     [column_constraint],
     ...
     [table_constraint][, ...]);
```

```
CREATE TABLE employees(
    employee_id    NUMBER(6),
    first_name     VARCHAR2(20),
    ...
    job_id         VARCHAR2(10) NOT NULL,
    CONSTRAINT emp_emp_id_pk
                   PRIMARY KEY (EMPLOYEE_ID));
```

# Defining Constraints

- Column constraint level

```
column [CONSTRAINT constraint_name] constraint_type,
```

- Table constraint level

```
column, ...  
  [CONSTRAINT constraint_name] constraint_type  
  (column, ...),
```


# The NOT NULL Constraint


Ensures that null values are not permitted for the column:


EMPLOYEE_ID	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	60
178	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10

■ ■ ■

20 rows selected.

  
**NOT NULL constraint**  
(No row can contain  
a null value for  
this column.)

  
**NOT NULL  
constraint**

  
**Absence of NOT NULL  
constraint**  
(Any row can contain  
null for this column.)

# The NOT NULL Constraint

Is defined at the column level:

```
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,  
    ...
```

← System  
named

← User  
named



# The UNIQUE Constraint

Defined at either the table level or the column level:

```
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 NOT NULL,  
    ...  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# The PRIMARY KEY Constraint

DEPARTMENTS

 PRIMARY KEY

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

...

Not allowed  
(Null value)



INSERT INTO

	Public Accounting		1400
50	Finance	124	1500

Not allowed  
(50 already exists)



# The PRIMARY KEY Constraint

Defined at either the table level or the column level:

```
CREATE TABLE departments(  
    department_id      NUMBER(4),  
    department_name    VARCHAR2(30)  
        CONSTRAINT dept_name_nn NOT NULL,  
    manager_id         NUMBER(6),  
    location_id        NUMBER(4),  
    CONSTRAINT dept_id_pk PRIMARY KEY(department_id));
```

# The FOREIGN KEY Constraint

## DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

**PRIMARY  
KEY**



...



## EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	60

**FOREIGN  
KEY**



...

**INSERT INTO**



200	Ford	9
201	Ford	60

**Not allowed  
(9 does not  
exist)**

**Allowed**



# The FOREIGN KEY Constraint

Defined at either the table level or the column level:

```
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 NOT NULL,  
    ...  
    department_id    NUMBER(4),  
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
        REFERENCES departments(department_id),  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# FOREIGN KEY Constraint

## Keywords

- **FOREIGN KEY:** Defines the column in the child table at the table constraint level
- **REFERENCES:** Identifies the table and column in the parent table
- **ON DELETE CASCADE:** Deletes the dependent rows in the child table when a row in the parent table is deleted.
- **ON DELETE SET NULL:** Converts dependent foreign key values to null

# The CHECK Constraint

- Defines a condition that each row must satisfy
- The following expressions are not allowed:
  - References to CURRVAL, NEXTVAL, LEVEL, and ROWNUM pseudocolumns
  - Calls to SYSDATE, UID, USER, and USERENV functions
  - Queries that refer to other values in other rows

```
..., salary    NUMBER(2)
      CONSTRAINT emp_salary_min
      CHECK (salary > 0),...
```

# Adding a Constraint Syntax

Use the **ALTER TABLE** statement to:

- Add or drop a constraint, but not modify its structure
- Enable or disable constraints
- Add a **NOT NULL** constraint by using the **MODIFY** clause

```
ALTER TABLE table  
ADD [CONSTRAINT constraint] type (column);
```



# Adding a Constraint

**Add a FOREIGN KEY constraint to the EMPLOYEES table indicating that a manager must already exist as a valid employee in the EMPLOYEES table.**

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

# Dropping a Constraint

- Remove the manager constraint from the EMPLOYEES table.

```
ALTER TABLE      employees
DROP CONSTRAINT    emp_manager_fk;
Table altered.
```

- Remove the PRIMARY KEY constraint on the DEPARTMENTS table and drop the associated FOREIGN KEY constraint on the EMPLOYEES.DEPARTMENT\_ID column.

```
ALTER TABLE      departments
DROP PRIMARY KEY CASCADE;
Table altered.
```

# Disabling Constraints

- Execute the **DISABLE** clause of the **ALTER TABLE** statement to deactivate an integrity constraint.
- Apply the **CASCADE** option to disable dependent integrity constraints.

```
ALTER TABLE          employees
DISABLE CONSTRAINT     emp_emp_id_pk CASCADE;
Table altered.
```

# Enabling Constraints

- Activate an integrity constraint currently disabled in the table definition by using the **ENABLE** clause.

```
ALTER TABLE      employees
ENABLE CONSTRAINT  emp_emp_id_pk;
Table altered.
```

- A **UNIQUE** or **PRIMARY KEY** index is automatically created if you enable a **UNIQUE** key or **PRIMARY KEY** constraint.

# Viewing Constraints

Query the `USER_CONSTRAINTS` table to view all constraint definitions and names.

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

CONSTRAINT_NAME	C	SEARCH_CONDITION
EMP_LAST_NAME_NN	C	"LAST_NAME" IS NOT NULL
EMP_EMAIL_NN	C	"EMAIL" IS NOT NULL
EMP_HIRE_DATE_NN	C	"HIRE_DATE" IS NOT NULL
EMP_JOB_NN	C	"JOB_ID" IS NOT NULL
EMP_SALARY_MIN	C	salary > 0
EMP_EMAIL_UK	U	

...

# Viewing the Columns Associated with Constraints

View the columns associated with the constraint names in the `USER_CONS_COLUMNS` view.

```
SELECT    constraint_name, column_name
FROM      user_cons_columns
WHERE     table_name = 'EMPLOYEES';
```

CONSTRAINT_NAME	COLUMN_NAME
EMP_DEPT_FK	DEPARTMENT_ID
EMP_EMAIL_NN	EMAIL
EMP_EMAIL_UK	EMAIL
EMP_EMP_ID_PK	EMPLOYEE_ID
EMP_HIRE_DATE_NN	HIRE_DATE
EMP_JOB_FK	JOB_ID
EMP_JOB_NN	JOB_ID

...

# Summary

**In this lesson, you should have learned how to create constraints.**

- **Types of constraints:**
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK
- **You can query the `USER_CONSTRAINTS` table to view all constraint definitions and names.**