

Lab 7: Including Constraints

Objectives

- ✓ Describe constraints
- ✓ Create and maintain constraints

Lesson Aim

In this lesson, you will learn how to implement business rules by including integrity 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 Description

NOT NULL specifies that the column cannot contain a null value

UNIQUE Specifies a column or combination of columns whose values must be unique for all rows in the table

PRIMARY KEY Uniquely identifies each row of the table

FOREIGN KEY Establishes and enforces a foreign key relationship between the column and a column of the referenced table

CHECK specifies a condition that must be true

The UNIQUE Constraint

A UNIQUE key integrity constraint requires that every value in a column or set of columns (key) be unique: that is, no two rows of a table can have duplicate values in a specified column or set of columns. The column (or set of columns) included in the definition of the UNIQUE key constraint is called the *unique key*. If the UNIQUE constraint comprises more than one column, that group of columns is called a *composite unique key*.

UNIQUE constraints allow the input of nulls for the columns. In fact, any number of rows can include nulls for columns.

Because nulls are not considered equal to anything, a null in a column (or in all columns of a composite UNIQUE key) always satisfies a UNIQUE constraint.

```
ALTER TABLE Persons
ADD CONSTRAINT uc_PersonID
UNIQUE (P_Id,LastName)
```

The PRIMARY KEY Constraint

A PRIMARY KEY constraint creates a primary key for the table. Only one primary key can be created for a each table. The PRIMARY KEY constraint is a column or set of columns that uniquely identifies each row in a table. This constraint enforces uniqueness of the column or column combination and ensures that no column that is part of the primary key can contain a null value.

```
ALTER TABLE Persons
ADD CONSTRAINT pk_PersonID
PRIMARY KEY (P_Id)
```

The FOREIGN KEY Constraint

The FOREIGN KEY, or referential integrity constraint, designates a column or combination of columns as a foreign key and establishes a relationship between a primary key or a unique key in the same table or a different table.

A foreign key value must match an existing value in the parent table or be NULL.

Foreign keys are based on data values and are purely logical, not physical, pointers.

```
ALTER TABLE Orders
ADD CONSTRAINT fk_PerOrders
FOREIGN KEY (P_Id)
REFERENCES Persons(P_Id)
```

The CHECK Constraint

The CHECK constraint defines a condition that each row must satisfy. A single column can have multiple CHECK constraints which reference the column in its definition. There is no limit to the number of CHECK constraints which you can define on a column.

```
ALTER TABLE Persons
ADD CONSTRAINT chk_Person
CHECK (P_Id>0 AND City='Sandnes')
```

Adding a Constraint Syntax

Use the ALTER TABLE statement to:

- Add or drop a constraint, but not modify its structure
- Add a NOT NULL constraint by using the MODIFY Clause

```
ALTER TABLE my_status
MODIFY ( person_id NOT NULL);
```

Dropping a Constraint

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

Remove the PRIMARY KEY constraint

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

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';
```

Oracle / PLSQL: Retrieve primary key information

```
SELECT cols.table_name, cols.column_name, cols.position, cons.status, cons.owner
FROM all_constraints cons, all_cons_columns cols
WHERE cons.constraint_type = 'P'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner
ORDER BY cols.table_name, cols.position;
```

Practice

1. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. The constraint should be named at creation. Name the constraint my_emp_id_pk. Hint: The constraint is enabled as soon as the ALTER TABLE command executes successfully.
2. Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my_deptid_pk. Hint: The constraint is enabled as soon as the ALTER TABLE command executes successfully.
3. Add a column DEPT_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my_emp_dept_id_fk.
4. Confirm that the constraints were added by querying the USER_CONSTRAINTS view. Note the types and names of the constraints. Save your statement text in a file called lab10_4.sql

CONSTRAINT_NAME	C
MY_DEPT_ID_PK	P
SYS_C002541	C
MY_EMP_ID_PK	P
MY_EMP_DEPT_ID_FK	R

5. Display the object names and types from the USER_OBJECTS data dictionary view for the EMP and DEPT tables. Notice that the new tables and a new index were created. If you have time, complete the following exercise:

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