

EXERCISE-3

INCLUDING CONSTRAINTS

OBJECTIVE

After the completion of this exercise the students should be able to do the following

- 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

The following types of integrity constraints are valid

a) **Domain Integrity**

- ✓ NOT NULL
- ✓ CHECK

b) **Entity Integrity**

- ✓ UNIQUE
- ✓ PRIMARY KEY

c) **Referential Integrity**

- ✓ FOREIGN KEY

Constraints can be created in either of two ways

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

Defining Constraints

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));

Domain Integrity

This constraint sets a range and any violations that takes place will prevent the user from performing the manipulation that caused the breach.It includes:

Example:

Assume table TEST1 with the following structure

```
CREATE TABLE test1 ( pk number PRIMARY KEY, fk number, col1 number,col2 number,  
CONSTRAINT fk_constraint FOREIGN KEY(fk) references test1, CONSTRAINT ck1 CHECK  
(pk>0 and col1>0), CONSTRAINT ck2 CHECK (col2>0));
```

An error is returned for the following statements

```
ALTER TABLE test1 DROP (pk);
```

```
ALTER TABLE test1 DROP (col1);
```

The above statement can be written with CASCADE CONSTRAINT

```
ALTER TABLE test 1 DROP(pk) CASCADE CONSTRAINTS;
```

(OR)

```
ALTER TABLE test 1 DROP(pk, fk, col1) CASCADE CONSTRAINTS;
```

VIEWING CONSTRAINTS

Query the USER_CONSTRAINTS table to view all the constraints definition and names.

Example:

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

Viewing the columns associated with constraints

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

Find the Solution for the following:

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.

```
ALTER TABLE emp ADD CONSTRAINT my-emp-id-pk PRIMARY KEY(id);
```

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_dept_id_pk.

```
ALTER TABLE dept ADD CONSTRAINT my-dept-id-pk PRIMARY KEY (id);
```

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 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-pk FOREIGN KEY (dept-id) REFERENCES dept(id);
```

4. 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.

```
ALTER TABLE emp ADD commission NUMBER (2,2);
```

```
ALTER TABLE emp ADD CONSTRAINT emp-commission-ck  
CHECK (commission > 0);
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	