

### 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';
```

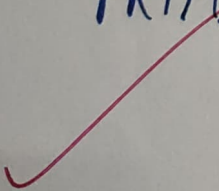
### Viewing the columns associated with constraints

```
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(7);  
ALTER TABLE EMP ADD CONSTRAINT my_emp_dept_id_fk  
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-com-check  
CHECK (COMMISSION > 0);
```

✓  
c. Pankh  
2/19/25

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