



**IPRC KARONGI**  
Integrated Polytechnic Regional College

P.O. Box 85 KARONGI- RWANDA  
Tel: +250 788871075  
Email: [info@iprcwest.ac.rw](mailto:info@iprcwest.ac.rw)  
[www.iprcakarongi.rp.ac.rw](http://www.iprcakarongi.rp.ac.rw)

---

**Department: INFORMATION AND COMMUNICATION TECHNOLOGY**

**Program: IT**

**Module code & name: ITLAD601 Database Administration**

**Credits: 9**

**Level: 6 Y2**

**Semester: 1**

**Academic year: 2023-2024**

**Module leader: TWIRINGIYIMANA Valens**

**Position: Assistant Lecturer**

## LEARNING UNIT 3 - MANAGE DATABASE

### 3.1 Maintaining of data integrity constraints

#### 3.1.1 Integrity Constraints

In Oracle, “constraints” are a facility to enforce rules to make sure that only allowable data values are stored in the database.

Integrity constraints provide a mechanism for ensuring that data conforms to guidelines specified by the database administrator. The most common types of constraints include:

**Entity Integrity:** It ensures that each row is uniquely identified in a table. To implement this mechanism, we use the primary key or unique constraint.

**Referential Integrity:** It ensures that to create a relationship between tables. To implement this mechanism, we use the foreign key (referential key).

**Domain integrity:** Domain is nothing but a column. It ensures that to check values with user define conditions before accepting values into a column. To perform this mechanism we use check, default, not null constraints.

#### 3.1.2 Enforcing Referential Integrity with Constraints

**Referential** Integrity constraint is enforced when a foreign key references the primary key of a relation. It specifies that all the values taken by the foreign key must either be available in the relation of the primary key or be null.

### 3.2 Methodical Creation of indexes and multiple-field keys according to business requirements

#### 3.2.1 Creating Indexes for Use with Constraints

##### a) NOT NULL

If we just add a column, by default the column is allowed to hold NULL values but in case there is a requirement that the column should not hold any NULL values. We can use NOT NULL constraint on that particular column. This will enforce the column to always have a value and it will not allow any NULL record to be added in the column. We will add NOT NULL constraint using both CREATE TABLE and ALTER TABLE constraint. We will create a table STUDENT with student\_id, LastName and firstName columns having NOT NULL constraint.

```
CREATE TABLE STUDENT ( student_id int NOT NULL, FirstName varchar(255) NOT NULL,  
LastName varchar(255) NOT NULL, Age int);
```

Let us now look at the ALTER statement query to add NOT NULL constraint to the age column of STUDENT TABLE.

```
ALTER TABLE STUDENT MODIFY AGE int NOT NULL;  
desc STUDENT;
```

## 2. UNIQUE

This constraint in Oracle ensures that all the values of the column are different from each other and there are no duplicates. We will again use two examples to understand. First using the CREATE statement and then using ALTER statement. We will create a table STUDENT with student\_id having UNIQUE constraint.

Let us look at the query using the CREATE TABLE statement.

```
CREATE TABLE student ( student_id int NOT NULL UNIQUE, FirstName varchar(25) NOT NULL, LastName varchar(25), Age int);
```

Now let add a UNIQUE constraint to age column using ALTER TABLE statement.

```
ALTER TABLE STUDENT ADD UNIQUE(age);
```

### 3. PRIMARY KEY

Primary key constraint uniquely describes each value of a column. No duplicates or NULL value is allowed. One important point is that a table can have only one primary key which in itself can be a combination of single or multiple fields. We will first create a table 'student' with student\_id as the primary key using the CREATE TABLE statement.

```
CREATE TABLE STUDENT ( student_id int PRIMARY KEY, FirstName varchar(255) NOT NULL, LastName varchar(255), Age int);
```

Let us now add primary Key using the ALTER TABLE statement to student\_id column after the STUDENT table has been created.

```
ALTER TABLE STUDENT ADD PRIMARY KEY (student_id);
```

### 4. FOREIGN KEY Constraints

A foreign [key](#) is a field which refers to the PRIMARY KEY of another table and the table which actually has the foreign key is called child table. Let us now create a table order which we has student\_id column as a foreign key referencing student\_id column of student table using CREATE TABLE statement.

```
CREATE TABLE Orders( OrderID int PRIMARY KEY, OrderNumber int NOT NULL, student_id int REFERENCES student(student_id));
```

Now we will use the ALTER TABLE statement to add a foreign key to the student\_id column of orders table.

```
ALTER TABLE Orders ADD FOREIGN KEY (student_id) REFERENCES student(student_id);
```

### 5. CHECK Constraint

The CHECK constraint is used to limit the value of the range that can be placed in a column. In case we want to restrict certain values in a column or a table we introduce the check constraint. We will introduce the CHECK constraint using both CREATE TABLE and ALTER TABLE statement. We are going to create a table student with age column having a check constraint.

```
CREATE TABLE student ( student_id int NOT NULL, FirstName varchar(25) NOT NULL, LastName varchar(25), Age int CHECK (Age>=18));
```

Let us now add CHECK constraint on the same column after the table student has been created using ALTER TABLE statement. ALTER TABLE student ADD CHECK (Age>=19);