4. Creating sample tables

* What is a constraint?
* Types of constraints?
* Sample tables
* Creating integrity constraints
* Creating example table
* Inserting sample data
* Summary
* Exercises

What is a constraint?

In the previous chapter we have seen how to create a table using CREATE TABLE command. Now we will understand how to define constraints. Constraints are used to implement standard and business rules. Data integrity of the database must be maintained. In order to ensure data has integrity we have to implement certain rules or constraints. As these constraints are used to maintain integrity they are called as integrity constraints.

Standard rules

Standard constraints are the rules related to primary key and foreign key. Every table must have a primary key. Primary key must be unique and not null. Foreign key must derive its values from corresponding parent key. These rules are universal and are called as standard rules.

Business rules

These rules are related to a single application. For example, in a payroll application we may have to implement a rule that prevents any row of an employee if salary of the employee is less than 2000. Another example is current balance of a bank account

Must be greater than or equal to 500.

Once the constraints are created, Oracle server makes sure that the constraints are not violated whenever a row is inserted, deleted or updated. If constraint is not satisfied then the operation will fail.

Constraints are normally defined at the time of creating table. But it is also possible to add constraints after the table is created using ALTER TABLE command. Constraints are stored in the Data Dictionary (a set of tables which stores information regarding database).

Each constraint has a name; it is either given by user using CONSTRAINT option or assigned by system. In the later case, the name is **SYS\_Cn**; where ***n*** is a number.

**Note:** It is recommended that you use constraint name so that referring to constraint will be easier later on.

Types of constraints

Constraints can be given at two different levels. If the constraint is related to a single column the constraint is given at the column level otherwise constraint is to be given at the table level. Base on the where a constraint is given, constraints are of two types:

* Column Constraints
* Table Constraints

Column Constraint

A constraint given at the column level is called as Column Constraint. It defines a rule for a single column. It cannot refer to column other than the column at which it is defined. A typical example is PRIMARY KEY constraint when a single column is the primary key of the table.

Table Constraint

A constraint given at the table level is called as Table Constraint. It may refer to more than one column of the table. A typical example is PRIMARY KEY constraint that is used to define composite primary key. A column level constraint can be given even at the table level, but a constraint that deals with more than one column must be given only at the table level.

The following is the syntax of CONSTRAINT clause used with CREATE TABLE and ALTER TABLE commands.

[CONSTRAINT constraint]

{ [NOT] NULL

| {UNIQUE | PRIMARY KEY}

| REFERENCES [schema.] table [(column)]

[ON DELETE CASCADE]

| CHECK (condition) }

The following is the syntax of table constraint.

[CONSTRAINT constraint]

{ {UNIQUE | PRIMARY KEY} (column [,column] ...)

| FOREIGN KEY (column [,column] ...)

REFERENCES [schema.] table [(column [,column] ...)]

[ON DELETE CASCADE]

| CHECK (condition)}

The main difference between column constraint and table constraint is that in table constraint we have to specify the name of the column for which the constraint is defined whereas in column constraint it is not required as constraint is given on immediately after the column.

Now let us understand sample table to be throughout this book. It is very important to understand these tables to get the best out of this book. I have made these tables to be easy to understand.

Sample tables

The following are the sample tables used throughout the book. These tables store information about course, batches and subject. There are six tables to store the required information by typical training center.

Let us first understand the meaning of each table.

The following are the required tables of our application.

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| Courses | Contains the details of all the courses offered by the institute. |
| Faculty | Contains the details of the faculty members of the institute. |
| Course\_faculty | This table contains information regarding which faculty can handle which course. It also contains rating regarding how good a faculty member is in handling a particular course. The rating is based on previous experience of the faulty member with that course. |
| Batches | Contains the information about all the batches. It contains information about all the batches that started and completed, on going and scheduled but not yet started. |
| Students | Contains information about all the students. Each student is assigned a new roll number whenever he/she joins a new course. |
| Payments | Information about all the payments made by students. A single student may pay course fee in multiple installments for a single course. |

**Table 1:** Sample tables.

The following few tables will give the list of columns of each of the table given in table 1.

COURSES Table

Contains information related to each course. Each course is given a unique code called course code.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CCODE | VARCHAR2(5) | Course Code. This is the primary key of the table. |
| NAME | VARCHAR(30) | Name of the course. |
| DURATION | NUMBER(3) | Duration of the course in no. of working days. |
| FEE | NUMBER(5) | Course fee of the course. |
| PREREQUISITE | VARCHAR2(100) | Prerequisite knowledge to do the course. |

The following are the required constraints of COURSES table.

* CCODE is primary key.
* FEE must be greater than or equal to 0.
* DURATION must be greater than or equal to 0.

FACULTY Table

Contains information about all the faculty members. Each faculty member is given a code called as FACCODE.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| FACCODE | VARCHAR2(5) | Faculty code. This is the primary key of the table. |
| NAME | VARCHAR2(30) | Name of the faculty. |
| QUAL | VARCHAR2(30) | Qualification of the faculty member. |
| EXP | VARCHAR2(100) | Experience of the faculty member. |

The following are the constraints of FACULTY table.

* FACCODE is primary key.

COURSE\_FACULTY table

Contains information regarding which faculty member can take which course. A single faculty member may be capable of handling multiple courses. However, each member is given a grade depending on his expertise in handling the subject. The grade will be wither A, B or C.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| FACCODE | VARCHAR2(5) | Faculty code. |
| CCODE | VARCHAR2(5) | Course the faculty can handle. |
| GRADE | CHAR(1) | Rating of faculty’s ability to handle this particular code.  A – Very good, B- Good,  C- Average. |

The following are the constraints of the table.

* FACCODE is a foreign key referencing FACCODE column of FACULTY table.
* CCODE is a foreign key referencing CCODE column of COURSES table.
* Primary key is consisting of FACCODE and CCODE.
* GRADE column must contain either A, B or C.

Batches table

Contains information about all the batches. These batches include batches that were completed, that are currently running and that are scheduled but yet to start.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| BCODE | VARCHAR2(5) | Code that is assigned to each batch. This is the primary key of the table. |
| CCODE | VARCHAR2(5) | Course code of the course of this batch. This is a foreign key referencing CCODE of COURSES table. |
| FACCODE | VARCHAR2(5) | Code of the faculty member taking this batch. |
| STDATE | DATE | Date on which the batch has started or scheduled to start if batch has not yet started. |
| ENDDATE | DATE | Date on which the batch has completed. If batch is not completed this will be null. |
| TIMING | NUMBER(1) | Number indicating the timing of the batch. 1- morning, 2 – after noon, and 3-evening. |

The following are the required constraints of this table.

* BCODE is the primary key.
* CCODE is a foreign key referencing CCODE of COURSES table.
* FACCODE is a foreign key referencing FACCODE of FACULTY table.
* STDATA must be <= ENDDATE
* TIMING column must be 1, 2 or 3.

STUDENTS table

Contains information about all the students of the institute. Each student is given a roll number. Roll number will be allotted to each student of each batch.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| ROLLNO | NUMBER(5) | Roll number that is assigned to each student. This is the primary key of the table. |
| BCODE | VARCHAR2(5) | Code of the batch to which student belongs. This is the foreign key referencing BCODE of BATCHES table. |
| NAME | VARCHAR2(30) | Name of the student. |
| GENDER | CHAR(1) | Gender of the student. M for male and F for female. |
| DJ | DATE | Date on which the student has joined. |
| PHONE | VARCHAR2(10) | Contact number of the student. |
| EMAIL | VARCHAR2(30) | Email address of the student. |

The following are the constraints of the table.

* ROLLNO is the primary key.
* BCODE is a foreign key referencing BCODE of BATCHES table.
* GENDER may be either M or F.

PAYMENTS table

Contains information about all the payment made by students of all bathes.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| ROLLNO | NUMBER(5) | Roll number of the student paying the fee. |
| DP | DATE | Date on which the amount is paid. |
| AMOUNT | NUMBER(5) | The amount paid by student. |

The following are the constraints.

* Primary key is consisting of ROLLNO and DP.
* AMOUNT must be >= 25