Database Systems Laboratory 3

Constraints

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

Constraints

Constraints

SQL Constraints

Constraints enforce rules on tables. Constraints can be imposed to the database tables either with the CREATE or ALTER command. Whenever a DML operation is to be performed on a table, the specified constraint must be satisfied for the operation to succeed

Naming a Constraint

A constraint can be identified by an internal or user-defined name. For a user's account, each constraint name must be unique. The standard convention for naming constraint is : **<table name>_<column name>_<constraint type>**

The abbreviation for different constraint types are: *pk* for PRIMARY Key, *fk* for FOREIGN Key, *uk* for UNIQUE, *chk* or *ck* for CHECK and *nn* for NOT NULL constraint
If you do not name a constraint, then the server will generate a name for it by using *SYS_Cn* format

Constraints...

Defining a Constraint

Constraint can be defined in either of the two ways:

- **Column level**:
  - A column- level constraint references a single column and is defined along with the definition of the column
  - This type of constraint is applied to the current column only
  - **column datatype [CONSTRAINT constraint_name] constraint_type**
- **Table level**:
  - A table- level constraint references one or more columns and is defined separately from the definitions of the columns
  - Except the NOT NULL constraint, all other constraints can be defined at the table level
  - **[CONSTRAINT constraint_name] constraint_type (column,..)**

Normally, simple keys are defined at the column level and composite keys are defined at the table level

NOT NULL Constraint

NOT NULL Constraint

It ensures that the column has a value and the value is not a NULL value
It prevents a column from accepting NULL values. The syntax is:
columnname datatype(size) NOT NULL or

columnname datatype(size) CONSTRAINT constraintname NOT NULL
It can only be applied at column level

name VARCHAR(20) CONSTRAINT student_name_nn NOT NULL

NOT NULL Constraint...

Let the structure of ITEM_MASTER table is:

Column	Type	Size
Item_no	NUMBER	4
Name	VARCHAR2	20
Qty_on_hand	NUMBER	5
Category	CHAR	1
Unit_measure	CHAR	4
Reorder_Lvl	NUMBER	5
Reorder_qty	NUMBER	5
Rate	NUMBER	8,2

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4), Name VARCHAR2(20), Qty_on_hand NUMBER(5), Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5), Reorder_qty NUMBER(5), Rate NUMBER(8,2));

Let the Item_no,Reorder_lvl, Reorder_qty and Rate columns are NOT NULL

NOT NULL Constraint...

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) NOT NULL, Name VARCHAR2(20), Qty_on_hand NUMBER(5), Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL);

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) CONSTRAINT c1 NOT NULL, Name VARCHAR2(20), Qty_on_hand NUMBER(5), Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) CONSTRAINT c2 NOT NULL, Reorder_qty NUMBER(5) CONSTRAINT c3 NOT NULL, Rate NUMBER(8,2) CONSTRAINT c4 NOT NULL);

Dropping NOT NULL Constraint

A NOT NULL constraint can be dropped by executing **ALTER TABLE tablename DROP CONSTRAINT constraintname;**

ALTER TABLE ITEM_MASTER DROP CONSTRAINT c4;

Unique Constraint

Unique Constraint

It ensures every value in a column or set of columns be unique
The unique constraint allows NULL values. The syntax is:

Column level: **Columnname datatype(size) UNIQUE** or

Columnname datatype(size) CONSTRAINT constraintname UNIQUE

Table level: **CONSTRAINT constraintname UNIQUE(columns)**

mob_no NUMBER(10) CONSTRAINT student_mob_uk UNIQUE

CONSTRAINT student_mob_uk UNIQUE(mob_no)

Unique Constraint...

Let the Name column in ITEM_MASTER table is unique:

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) NOT NULL, Name VARCHAR2(20) UNIQUE, Qty_on_hand NUMBER(5), Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL);

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) NOT NULL, Name VARCHAR2(20) , Qty_on_hand NUMBER(5), Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL), CONSTRAINT ce3 UNIQUE(Name) ;

Unique Constraint...

The composite unique key constraint can be defined only at the table level by specifying column names separated by a comma within parentheses

CONSTRAINT student_name_city_uk UNIQUE(name, city)

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) NOT NULL, Name VARCHAR2(20) , Qty_on_hand NUMBER(5), Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL), CONSTRAINT ce4 UNIQUE(Item_no,Name) ;

Dealing with UNIQUE Constraint in an existing table

The syntax for adding unique constraint is:
ALTER TABLE tablename ADD CONSTRAINT constraintname UNIQUE(columns);

ALTER TABLE ITEM_MASTER ADD CONSTRAINT C4 UNIQUE(Name);

The syntax for dropping unique constraint is:
ALTER TABLE tablename DROP CONSTRAINT constraintname;

ALTER TABLE ITEM_MASTER DROP CONSTRAINT C4;

Primary Key Constraint

Primary Key Constraint

Primary key constraint is also known as the entity integrity constraint
A table can have at most one primary key constraint
PRIMARY key is equivalent to the combination of NOT NULL constraint and UNIQUE constraint

Column level: **Columnname datatype(size) PRIMARY KEY** or

Columnname datatype(size) CONSTRAINT constraintname PRIMARY KEY

Table level: **CONSTRAINT constraintname PRIMARY KEY(columns)**

roll number(6) CONSTRAINT student_roll_pk PRIMARY KEY

CONSTRAINT student_roll_pk PRIMARY KEY(roll)

Primary Key Constraint...

Let the Item_no column in ITEM_MASTER table is primary key:

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) PRIMARY KEY, Name VARCHAR2(20) UNIQUE, Qty_on_hand NUMBER(5), Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL);

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4), Name VARCHAR2(20) UNIQUE, Qty_on_hand NUMBER(5), Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL, CONSTRAINT C7 PRIMARY KEY(Item_no));

Dealing with Primary Key Constraint in an existing table

The syntax for adding Primary key constraint is:
ALTER TABLE tablename ADD CONSTRAINT constraintname PRIMARY KEY(columns);

ALTER TABLE ITEM_MASTER ADD CONSTRAINT C5 PRIMARY KEY(Item_no);

The syntax for dropping Primary key constraint is:
ALTER TABLE tablename DROP PRIMARY KEY [CASCADE];

ALTER TABLE ITEM_MASTER DROP PRIMARY KEY; or

ALTER TABLE ITEM_MASTER DROP CONSTRAINT C5;

Foreign Key Constraint

Foreign Key Constraint

It is also known as the referential integrity constraint. It establishes a relationship with the primary key of the same or another table. Foreign key and the referenced primary key columns need not have the same name, but the data type, size and domain must match

Column level: **Columnname datatype(size) [CONSTRAINT constraintname] REFERENCES tablename(columns)** or **Columnname datatype(size) [CONSTRAINT constraintname] REFERENCES tablename**

Table level: **CONSTRAINT constraintname FOREIGN KEY(columns) REFERENCES tablename(columns)**

fid VARCHAR(6) CONSTRAINT student_fid_fk REFERENCES faculty(fid)

CONSTRAINT student_fid_fk FOREIGN KEY(fid) REFERENCES faculty(fid)

Foreign Key Constraint...

ON DELETE CASCADE

This option can be added to allow deletion of a record in the parent table and deletion of the dependent records in the child table implicitly

Coulmn level: **Columnname datatype(size) [CONSTRAINT constraintname] REFERENCES tablename(columns) [ON DELETE CASCADE]**

Table level: **CONSTRAINT constraintname FOREIGN KEY(columns) REFERENCES tablename(columns) [ON DELETE CASCADE]**

fid VARCHAR(6) CONSTRAINT student_fid_fk REFERENCES faculty(fid) ON DELETE CASCADE

CONSTRAINT student_fid_fk FOREIGN KEY(fid) REFERENCES faculty(fid) ON DELETE CASCADE

Foreign Key Constraint...

Let ITEM_TRANS be the table where It_no references to the Item_no column in ITEM_MASTER table

Column	Type	Size
It_no	NUMBER	4
Trans_date	DATE	
qty	NUMBER	5

CREATE TABLE ITEM_TRANS(It_no NUMBER(4) REFERENCES ITEM_MASTER(Item_no), trans_date DATE, qty NUMBER(5));

Dealing with Foreign Key Constraint in an existing table

The syntax for adding Foreign key constraint is:
ALTER TABLE tablename ADD CONSTRAINT constraintname FOREIGN KEY(columns) REFERENCES tablename(columns);

ALTER TABLE ITEM_TRANS ADD CONSTRAINT C7 FOREIGN KEY(Item_no) REFERENCES ITEM_MASTER(Item_no);

The syntax for dropping Foreign key constraint is:
ALTER TABLE tablename DROP CONSTRAINT constraintname;

ALTER TABLE ITEM_TRANS DROP CONSTRAINT C7;

Check Constraint

Check Constraint

It defines a condition that every row must satisfy. There can be more than one CHECK constraint on a column

Column level: **Columnname datatype(size) CONSTRAINT constraintname CHECK(condition)**

Table level: **CONSTRAINT constraintname CHECK(condition)**

age NUMBER(2) CONSTRAINT student_age_chk CHECK((age>=15) AND (age<=50))

CONSTRAINT student_age_chk CHECK((age>=15) AND (age<=50))

name VARCHAR(20) CONSTRAINT student_name_nn CHECK(name is NOT NULL)

Check Constraint...

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) PRIMARY KEY, Name VARCHAR2(20) UNIQUE, Qty_on_hand NUMBER(5), Category CHAR(1) CHECK(Category in('A', 'B', 'C'), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL);

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) PRIMARY KEY, Name VARCHAR2(20) UNIQUE, Qty_on_hand NUMBER(5), Category CHAR(1) NOT NULL, Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL, CHECK((Category='A' AND Rate<=1000) OR (Category='B' AND Rate<=4500) OR (Category='C' AND Rate>=4500)));

Dealing with Check Constraint in an existing table

The syntax for adding Check constraint is:
ALTER TABLE tablename ADD CONSTRAINT constraintname CHECK (condition);

ALTER TABLE ITEM_TRANS ADD CONSTRAINT C8 CHECK(Category in('A', 'B', 'C'));

The syntax for dropping Check constraint is:
ALTER TABLE tablename DROP CONSTRAINT constraintname;

ALTER TABLE ITEM_TRANS DROP CONSTRAINT C8;

DEFAULT Value

DEFAULT Value

It ensures that a particular column will always have a value when a new record is inserted. The default value gets overwritten if a user enters another value. The default value is used if a NULL value is inserted. The DEFAULT value is defined in the column level. The syntax is:

Columnname datatype(size) DEFAULT value

CREATE TABLE ITEM_MASTER(Item_no NUMBER(4) PRIMARY KEY, Name VARCHAR2(20) UNIQUE, Qty_on_hand NUMBER(5) DEFAULT 100, Category CHAR(1), Unit_measure CHAR(4), Reorder_Lvl NUMBER(5) NOT NULL, Reorder_qty NUMBER(5) NOT NULL, Rate NUMBER(8,2) NOT NULL);

DEFAULT Value...

If a column level constraint is defined on the column with a default value, then the default value must precede the constraint. The syntax is:

Columnname datatype(size) DEFAULT value constraint definition

Qty_on_hand NUMBER(5) DEFAULT 100 CHECK (Qty_on_hand>=100),

Viewing USER Constraints

Viewing USER Constraints

User can view all the constraints by excuting
SELECT * FROM USER_CONSTRAINTS;

If the user wants to view all the constraints applied to a single table, the syntax is:
SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME= tablename;

*SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME='ITEM_MASTER';*