
ORACLE
ACADEMY

Database Programming with SQL

14-2
PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

ORACLE ACADEMY

Objectives

This lesson covers the following objectives:

- Define and give an example of a PRIMARY KEY, FOREIGN KEY, and CHECK constraint
- Explain the purpose of defining PRIMARY KEY, FOREIGN KEY, and CHECK constraints
- Demonstrate the creation of constraints at the column level and table level in a CREATE TABLE statement
- Evaluate a business problem requiring the addition of a PRIMARY KEY and FOREIGN KEY constraint and write the code to execute the change

Purpose

- As discussed in the last section, constraints are used to prevent invalid data entry into database tables.
- What would happen if, surreptitiously or just through a careless mistake, your personal unique identification was given to another person?
- What if tomorrow at school someone else was credited with your classes for graduation or was able to eat lunch using your lunch-card number?
- Ensuring data integrity is what constraints are all about. After all, you're unique!

PRIMARY KEY Constraints

- A PRIMARY KEY constraint is a rule that the values in one column or a combination of columns must uniquely identify each row in a table.
- No primary-key value can appear in more than one row in the table.
- To satisfy a PRIMARY KEY constraint, both of the following conditions must be true:
  - No column that is part of the primary key can contain a null.
  - A table can have only one primary key.

PRIMARY KEY Constraints

- PRIMARY KEY constraints can be defined at the column or the table level.
- However, if a composite PRIMARY KEY is created, it must be defined at the table level.
- When defining PRIMARY KEY columns, it is a good practice to use the suffix _pk in the constraint name.
- For example, the constraint name for the PRIMARY KEY column named client_number in table named CLIENTS could be clients_client_num_pk.

PRIMARY KEY Constraints

- In a CREATE TABLE statement, the column-level PRIMARY KEY constraint syntax is stated:

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
first_name VARCHAR2(14),
last_name VARCHAR2(13));
```

- Note that the column-level simply refers to the area in the CREATE TABLE statement where the columns are defined.
- The table level refers to the last line in the statement below the list of individual column names.

PRIMARY KEY Constraints

- To create the PRIMARY KEY constraint at table-level the syntax is:

```
CREATE TABLE clients
(client_number NUMBER(4),
 first_name VARCHAR2(14),
 last_name VARCHAR2(13),
CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

- Note that the PRIMARY KEY column name follows the constraint type, and is enclosed in parenthesis.

PRIMARY KEY Constraints

- To define a composite PRIMARY KEY, you must define the constraint at the table level rather than the column level.
- An example of a composite primary key constraint is shown below:

```
CREATE TABLE copy_job_history
(employee_id NUMBER(6,0),
 start_date DATE,
 job_id VARCHAR2(10),
 department_id NUMBER(4,0),
CONSTRAINT copy_jhist_id_st_date_pk PRIMARY KEY(employee_id, start_date));
```

FOREIGN KEY (REFERENTIAL INTEGRITY) Constraints

- FOREIGN KEY constraints are also called "referential integrity" constraints.
- Foreign Key constraints designate a column or combination of columns as a foreign key.
- A foreign keys links back to the primary key (or a unique key) in another table, and this link is the basis of the relationship between tables.

Viewing a Foreign Key

- The table containing the foreign key is called the "child" table and the table containing the referenced key is called the "parent" table.

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Viewing a Foreign Key

- In the tables shown, the primary-key of the DEPARTMENTS table, department_id, also appears in the EMPLOYEES table as a foreign-key column.

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Referential-integrity Constraint

- To satisfy a referential-integrity constraint, a foreign-key value must match an existing value in the parent table or be NULL.

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Referential-integrity Constraint

- A primary-key value can exist without a corresponding foreign-key value; however, a foreign-key must have a corresponding primary key.

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Referential-Integrity Constraint Rule

- The rule is: before you define a referential-integrity constraint in the child table, the referenced UNIQUE or PRIMARY KEY constraint on the parent table must already be defined.

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Referential-Integrity Constraint Rule

- In other words, you must first have a parent primary key defined before you can create a foreign key in a child table.

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

FOREIGN KEY Constraint

- To define a FOREIGN KEY constraint, it is good practice to use the suffix _fk in the constraint name.
- For example, the constraint name for the FOREIGN KEY column department_id in the employees table could be named emps_dept_id_fk.

FOREIGN KEY Constraint Syntax

- The syntax for defining a FOREIGN KEY constraint requires a reference to the table and column in the parent table.
- A FOREIGN KEY constraint in a CREATE TABLE statement can be defined as follows.
- Column-level syntax example:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0) CONSTRAINT c_emps_dept_id_fk
                           REFERENCES departments(department_id),
 email VARCHAR2(25));
```

FOREIGN KEY Constraint Syntax

- The syntax for defining a FOREIGN KEY constraint requires a reference to the table and column in the parent table.
- A FOREIGN KEY constraint in a CREATE TABLE statement can be defined as follows.
- Table-level syntax example:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT c_emps_dept_id_fk FOREIGN KEY (department_id)
                              REFERENCES departments(department_id));
```

ON DELETE CASCADE - Maintaining Referential Integrity

- Using the ON DELETE CASCADE option when defining a foreign key enables the dependent rows in the child table to be deleted when a row in the parent table is deleted.
- If the foreign key does not have an ON DELETE CASCADE option, referenced rows in the parent table cannot be deleted.
- In other words, the child table FOREIGN KEY constraint includes the ON DELETE CASCADE permission allowing its parent to delete the rows that it refers to.

ON DELETE CASCADE - Maintaining Referential Integrity

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

ON DELETE CASCADE

- If the department_id column in employees was created with the ON DELETE CASCADE option specified, the DELETE statement issued on the departments table will execute.
- If the ON DELETE CASCADE option was not specified when the FOREIGN KEY was created, an attempt to delete a department from the departments table that has entries in the employees table will fail.

ON DELETE CASCADE Syntax

- Table created without ON DELETE CASCADE:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
        REFERENCES copy_departments(department_id));
```

- An attempt to delete department_id 110 from the departments table fails as there are dependent rows in the employee table.

```
ORA-02292: integrity constraint (US_A009EMEA815_PLSQL_T01.CDEPT_DEPT_ID_FK)
violated - child record found
```

ON DELETE CASCADE Syntax

- Table created with ON DELETE CASCADE:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
    REFERENCES copy_departments(department_id) ON DELETE CASCADE);
```

- An attempt to delete department_id 110 from the departments table succeeds, and the dependent rows in the employee table are also deleted.
- 1 row(s) deleted.

ON DELETE SET NULL

- Rather than having the rows in the child table deleted when using an ON DELETE CASCADE option, the child rows can be filled with null values using the ON DELETE SET NULL option.

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
    REFERENCES copy_departments(department_id) ON DELETE SET NULL);
```

ON DELETE SET NULL

- This could be useful when the parent table value is being changed to a new number such as converting inventory numbers to bar-code numbers.
- You would not want to delete the rows in the child table.
- When the new bar-code numbers are entered into the parent table, they would then be able to be inserted into the child table without having to totally re-create each child table row.

CHECK Constraints

- The CHECK constraint explicitly defines a condition that must be met.
- To satisfy the constraint, each row in the table must make the condition either True or unknown (due to a null).
- The condition of a CHECK constraint can refer to any column in the specified table, but not to columns of other tables.

CHECK Constraint Example

- This CHECK constraint ensures that a value entered for end_date is later than start_date.

```
CREATE TABLE copy_job_history
(employee_id NUMBER(6,0),
 start_date DATE,
 end_date DATE,
 job_id VARCHAR2(10),
 department_id NUMBER(4,0),
 CONSTRAINT cjhist_emp_id_st_date_pk
           PRIMARY KEY(employee_id, start_date),
 CONSTRAINT cjhist_end_ck CHECK (end_date > start_date));
```

- As this CHECK CONSTRAINT is referencing two columns in the table, it MUST be defined at table level.

CHECK Constraint Conditions

- A CHECK constraint must only be on the row where the constraint is defined.
- A CHECK constraint cannot be used in queries that refer to values in other rows.
- The CHECK constraint cannot contain calls to the functions SYSDATE, UID, USER, or USERENV.
- The statement CHECK(SYSDATE >'05-May-1999') is not allowed.

CHECK Constraint Conditions

- The CHECK constraint cannot use the pseudocolumns CURRVAL, NEXTVAL, LEVEL, or ROWNUM.
- The statement CHECK(NEXTVAL > 0) is not allowed.
- A single column can have multiple CHECK constraints that reference the column in its definition.
- There is no limit to the number of CHECK constraints that you can define on a column.

CHECK Constraint Syntax

- CHECK constraints can be defined at the column level or the table level.
- The syntax to define a CHECK constraint is:
  - Column-level syntax:

```
salary NUMBER(8,2) CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```

  - Table-level syntax:

```
CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```

Terminology

Key terms used in this lesson included:

- CHECK constraint
- FOREIGN KEY constraint
- REFERENCES
- NOT NULL
- ON DELETE CASCADE
- ON DELETE SET NULL
- PRIMARY KEY constraint

Summary

In this lesson, you should have learned how to:

- Define and give an example of a PRIMARY KEY, FOREIGN KEY, and CHECK constraint
- Explain the purpose of defining PRIMARY KEY, FOREIGN KEY, and CHECK constraints
- Demonstrate the creation of constraints at the column level and table level in a CREATE TABLE statement
- Evaluate a business problem requiring the addition of a PRIMARY KEY and FOREIGN KEY constraint and write the code to execute the change


ORACLE
ACADEMY