

SQL

Constraints

dr Szymon Murawski

Comarch SA

December 12, 2019

CHECK constraints

```
CREATE TABLE students (  
    pesel CHAR(11) CHECK(LENGTH(PESEL) = 11),  
    first_name VARCHAR(300),  
    last_name VARCHAR(300) NOT NULL,  
    birthdate DATE,  
    matriculation_date DATE,  
    CHECK (matriculation_date > birthdate)  
);
```

Every time the row changes (via insert, update or delete) database will check if all the check constraints are fulfilled by new value. If not, then the operation will be rolled back.

Inserting new rows

```
INSERT INTO students(pesel, birthdate, matriculation_date)
VALUES ('39090100', '1939-09-01', '2005-10-01')
```

```
ERROR:  new row for relation "students" violates check constraint
        "students_pesel_check"
```

```
INSERT INTO students
VALUES ('39090100001', '2005-10-01', '2005-09-01')
```

```
ERROR:  new row for relation "students" violates check constraint
        "students_check"
```

```
INSERT INTO students
VALUES ('39090100001', '1939-09-01', '2005-10-01')
```

Query returned successfully: one row affected

NOT NULL constraints

```
CREATE TABLE students (  
    pesel CHAR(11) CHECK(LENGTH(pesel) = 11),  
    first_name VARCHAR(300),  
    last_name VARCHAR(300) NOT NULL,  
    birthdate DATE,  
    matriculation_date DATE,  
    CHECK (matriculation_date > birthdate)  
);
```

Every time the row changes (via insert, update or delete) database will check if the value is not null. If not, then the operation will be rolled back.

UNIQUE constraints

```
CREATE TABLE students (  
    pesel CHAR(11) CHECK(LENGTH(pesel) = 11) UNIQUE,  
    first_name VARCHAR(300),  
    last_name VARCHAR(300) NOT NULL,  
    birthdate DATE,  
    matriculation_date DATE,  
    CHECK (matriculation_date > birthdate),  
    UNIQUE(first_name, last_name)  
);
```

Every time the row changes (via insert, update or delete) database will check if the value is unique in the column. If not, then the operation will be rolled back.

PRIMARY KEY constraint

```
CREATE TABLE students (  
    pesel CHAR(11) CHECK(LENGTH(pesel) = 11) PRIMARY KEY,  
    first_name VARCHAR(300),  
    last_name VARCHAR(300) NOT NULL,  
    birthdate DATE,  
    matriculation_date DATE,  
    CHECK (matriculation_date > birthdate),  
    UNIQUE(first_name, last_name)  
);
```

Primary key uniquely identifies row in a table. Column (or columns) that are primary key will also have NOT NULL and UNIQUE constraints placed on them.

REFERENCE constraints

```
CREATE TABLE grades (  
    student_pesel CHAR(11) REFERENCES students,  
    grade VARCHAR,  
    date DATE,  
    course VARCHAR(100)  
);
```

Every time the row changes (via insert, update or delete) database will check if the value in column "pesel" exists as a primary key in table "students"

REFERENCE constraints

```
CREATE TABLE grades (  
    student_pesel CHAR(11) REFERENCES students(pesel),  
    grade VARCHAR,  
    date DATE,  
    course_id INTEGER,  
    FOREIGN KEY (course_id) REFERENCES courses(id)  
);
```

Every time the row changes (via insert, update or delete) database will check if the value in column "student_pesel" exists in column "pesel" in table "students", as well as whether "course_id" exists in "courses.id".

REFERENCE constraints actions

```
CREATE TABLE grades (  
    student_pesel CHAR(11) REFERENCES students(pesel) ON DELETE  
        CASCADE,  
    grade VARCHAR,  
    date DATE,  
    course VARCHAR(100),  
    FOREIGN KEY (course_id) REFERENCES courses(id) ON UPDATE  
        CASCADE ON DELETE SET NULL  
);
```

If row in the referenced table changes, those changes are propagated (cascaded) down. In this case if we delete a specific student we will delete all of grades connected to that student. On deleting a row from "courses" relation we will set to NULL all connected rows in grades and on updating a row in "courses" we will cascade that update.

Naming constraints

```
CREATE TABLE grades (  
    student_pesel CHAR(11) REFERENCES students(pesel) ON DELETE  
        CASCADE,  
    grade VARCHAR,  
    date DATE,  
    course VARCHAR(100),  
    CONSTRAINT students_course_id_fk FOREIGN KEY (course_id)  
        REFERENCES courses(id) ON UPDATE CASCADE ON DELETE SET NULL,  
    CONSTRAINT grade_check CHECK (grade IN ('A', 'B', 'C', 'D', 'E'  
        , 'F'))  
);
```

Constraints can be named only, when they are declared as "standalone", not during declaration of a column. If the name is not provided, database will automatically generate one.

Displaying constraints

```
SELECT tc.constraint_name, tc.table_name, kcu.column_name,
       constraint_type
FROM
  information_schema.table_constraints tc
JOIN information_schema.key_column_usage kcu
  ON tc.constraint_name = kcu.constraint_name;
```

constraint_name	table_name	column_name	type
clients_pkey	clients	client_id	PRIMARY KEY
actors_pkey	actors	actor_id	PRIMARY KEY
starring_movie_id_fkey	starring	movie_id	FOREIGN KEY
starring_actor_id_fkey	starring	actor_id	FOREIGN KEY
starring_pkey	starring	actor_id	PRIMARY KEY
starring_pkey	starring	movie_id	PRIMARY KEY

(0 rows)

Dropping constraints

```
CREATE TABLE grades (  
    student_pesel CHAR(11) REFERENCES students(pesel) ON DELETE  
        CASCADE,  
    grade VARCHAR,  
    date DATE,  
    course VARCHAR(100),  
    CONSTRAINT students_course_id_fk FOREIGN KEY (course_id)  
        REFERENCES courses(id) ON UPDATE CASCADE ON DELETE SET NULL,  
    CONSTRAINT grade_check CHECK (grade IN ('A', 'B', 'C', 'D', 'E'  
        , 'F'))  
);  
ALTER TABLE grades DROP CONSTRAINT grade_check;
```

Constraints can be dropped like any other database element. We need to provide a name for the constraint to drop.

Adding new constraints

```
CREATE TABLE grades (  
    student_pesel CHAR(11) REFERENCES students(pesel) ON DELETE  
        CASCADE,  
    grade VARCHAR,  
    date DATE,  
    course VARCHAR(100),  
    CONSTRAINT students_course_id_fk FOREIGN KEY (course_id)  
        REFERENCES courses(id) ON UPDATE CASCADE ON DELETE SET NULL,  
    CONSTRAINT grade_check CHECK (grade IN ('A', 'B', 'C', 'D', 'E'  
        , 'F'))  
);  
ALTER TABLE grades DROP CONSTRAINT students_course_id_fk;  
ALTER TABLE grades ADD CONSTRAINT students_course_id_fk FOREIGN  
    KEY (course_id) REFERENCES courses(id) ON UPDATE CASCADE ON  
    DELETE CASCADE
```

Constraints can be placed upon tables by ALTER TABLE command.