

## Introduction to Rational Database in SQL

笔记本: 学习

创建时间: 2020/2/13 17:11

更新时间: 2020/2/15 16:11

作者: マーク先生

URL: <https://campus.datacamp.com/courses/introduction-to-relational-databases-in-...>

# Introduction to Rational Database in SQL

## Your First Database

- Database Constraint

约束是数据库模式定义的一部分。

约束通常与一个表相关联, 并使用 `CREATE CONSTRAINT` 或 `CREATE ASSERTION` SQL语句创建。

他们定义数据库中的数据必须符合的某些属性。他们可以应用于列, 整个表格, 多个表格或整个模式。可靠的数据库系统可以确保约束始终保持不变 (除了可能在事务内部, 对于所谓的延迟约束)。

常见的约束条件是:

- **not null** - 列中的每个值都不能为 `NULL`
- **unique** - value (s) 对于表中的每一行必须是唯一的
- **primary key** - 指定列中的值对于表中的每一行必须是唯一的, 而不是 `NULL`; 通常数据库中的每个表都应该有一个主键 - 它用于识别单个记录
- 指定列中的 **foreign key** 必须引用另一个表中的现有记录 (通过它的主键或其他唯一约束)
- **check** - 指定一个表达式, 为了满足约束条件, 它必须计算为真

- Database is a group of table with columns, we have several instructions and several code for database.
- Remember to put a ; after each sentences

```
CREATE TABLE students (  
  last_name varchar(128) NOT NULL,  
  ssn integer PRIMARY KEY,  
  phone_no char(12)  
);  
高级创造table
```

命令	解释
CREATE TABLE xxx (column datatype, column datatype...)	How to create a table
ALTER TABLE xxx ADD COLUMN column datatype	add a column in the table
ALTER TABLE xxxx RENAME COLUMN aaa TO bbb	Rename Columns
ALTER TABLE xx DROP COLUMN aa	Drop a column
INSERT INTO xxx SELECT DISTINCT xxx FROM xx	Insert into the table using existing table
Drop TABLE xxx	drop table
ALTER TABLE xx DROP COLUMN aaa	Drop column from table

# Enforce data consistency with attribute constraints

- Constraints give the data structure
- Constraints help with consistency, and thus data quality
- Data quality is a business advantage / data science prerequisite
- Enforcing is difficult, but PostgreSQL helps

## Datatype

- text: character strings of any length
- varchar [ (x) ]: a maximum of n characters
- char [ (x) ]: a fixed-length string of n characters
- boolean: can only take three states, e.g. TRUE, FALSE and NULL (unknown)
- date, time and timestamp: various formats for date and time calculations
- numeric: arbitrary precision numbers, e.g. 3.1457
- integer: whole numbers in the range of -2147483648 and +2147483647

- NULL!=NULL 和 NOT NULL  
null有可能是不存在的意思，也有可能是数据缺失。设定not null就是令这个东

西一定得存在。就是防止数据缺失。当然如果数据可以以不存在的形式存在，就可以不用设置not null。

命令	解释
INSERT INTO xxx (a,b,c) VALUES (1,2,3)	add a row inside
SELECT CAST(xx AS datatype) FROM xx	change datatype in select
ALTER TABLE xx ALTER COLUMN xxx TYPE xxx	change types
ALTER TABLE xx ALTER COLUMN xxx TYPE xxx USING SUBSTRING (col FROM 1 FOR a)	IF too long to convert, use substring to convert it to certain number
ALTER TABLE xxx ALTER COLUMN xxx SET NOT NULL;	change a column to set it to NOT NULL
ALTER TABLE x ADD CONSTRAINT aaa UNIQUE(bb)	add a new column aaa which is the unique one of column bb

## Uniquely identify records with key constraints

- **Superkey and candidate key**

1. Count the distinct records for all possible combinations of columns. If the resulting number  $x$  equals the number of all rows in the table for a combination, you have discovered a superkey. 相当于distinct等于所有的组合就是superkey。
2. Then remove one column after another until you can no longer remove columns without seeing the number  $x$  decrease. If that is the case, you have discovered a (candidate) key.
3. Primary Key: Ideally distinct key
4. Surrogate Key: A surrogate key is any column or set of columns that can be declared as the primary key instead of a "real" or natural key

### Super Key

**Super Key** is defined as a set of attributes within a table that can uniquely identify each record within a table. Super Key is a superset of Candidate key.

In the table defined above super key would include `student_id`, `(student_id, name)`, `phone` etc.

Confused? The first one is pretty simple as `student_id` is unique for every row of data, hence it can be used to identity each row uniquely.

Next comes, `(student_id, name)`, now name of two students can be same, but their `student_id` can't be same hence this combination can also be a key.

## Candidate Key

Candidate keys are defined as the minimal set of fields which can uniquely identify each record in a table. It is an attribute or a set of attributes that can act as a Primary Key for a table to uniquely identify each record in that table. There can be more than one candidate key.


In our example, `student_id` and `phone` both are candidate keys for table **Student**.

- A candidate key can never be NULL or empty. And its value should be unique.
- There can be more than one candidate keys for a table.
- A candidate key can be a combination of more than one columns(attributes).

## Primary Key

Primary key is a candidate key that is most appropriate to become the main key for any table. It is a key that can uniquely identify each record in a table.

Primary Key for this table




student_id	name	age	phone

For the table **Student** we can make the `student_id` column as the primary key.

## Composite Key

Key that consists of two or more attributes that uniquely identify any record in a table is called **Composite key**. But the attributes which together form the **Composite key** are not a key independently or individually.

Composite Key



student_id	subject_id	marks	exam_name

Score Table - To save scores of the student for various subjects.

## Secondary or Alternative key

The candidate key which are not selected as primary key are known as secondary keys or alternative keys.

## Non-key Attributes

**Non-key** attributes are the attributes or fields of a table, other than **candidate key** attributes/fields in a table.

## Non-prime Attributes

**Non-prime** Attributes are attributes other than **Primary Key attribute(s)**.

命令	解释
SELECT COUNT(DISTINCT(col)) FROM xx	count the unique row in certain column
ALTER TABLE xx ADD CONSTRAINT a PRIMARY KEY (b);	let b to be primary key and rename it to a
ALTER TABLE a ADD COLUMN b serial	Serial is a special type, which can do auto-increment stuff, add it will auto generate 1-N to a column, it can be treated as surrogate key
UPDATE a SET b = CONCAT(c, d);	Update table a, set new column be as the result of old columns c+d
高级方法创造table	在第一节里面

## Glue together tables with foreign keys

- Foreign Key: Definition - What does Foreign Key mean? A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables because it references the primary key of another table, thereby establishing a link between them. 记住，只有primary key. 而且作为foreign key不会接受新的东西，所以如果要加入新的row，这一列只能是已知的key.
- Integrity Violations  
If one column is a foreign key, delete it or change it might gives a error.
- Dealing with violation  
ON DELETE...
  - ...NO ACTION: Throw an error
  - ...CASCADE: Delete all referencing records
  - ...RESTRICT: Throw an error
  - ...SET NULL: Set the referencing column to NULL
  - ...SET DEFAULT: Set the referencing column to its default value

命令	解释
----	----

命令	解释
ALTER TABLE a ADD CONSTRAINT b FOREIGN KEY (c) REFERENCES d (e)	In table a choose a column c and let it to be a foreign key refer to the column e in table d and rename to a
ALTER TABLE a ADD CONSTRAINT b FOREIGN KEY (c) REFERENCES d (e) ON DELETE CASCADE	重点在于最后一段，如果是ON DELETE NO ACTION, 则会变得无法删除其中一方，如果是ON DELETE CASCADE, 就可以在删除一方时两个一起删除，避免了无法删除的现象

