

# LU06.A03 - SQL-DDL: Constraint Management

## Requirements

- Work type: individual
- Timeframe: 30 Minutes
- Means of aid:
  - Only teaching materials, no websearch, no use of ai.
  - [W3Schools](#) | [SQL Editor](#)
- Expected result: Semantically and syntactically correct SQL statements according to the requirements of the case studies.

## Case studies / Assignments

Here are five assignments, each covering a specific MySQL constraint. As usual, along with the solutions at the end (link).

## Assignments

### A: PRIMARY KEY

Create a table books that has a book\_id as a unique identifier for each book, with book\_id as the primary key. Include columns for book\_title (VARCHAR) and author\_name (VARCHAR). The book\_id should be an integer and cannot be NULL.

### B: AUTO INCREMENT

Create a table products where each product has an automatically generated, unique product\_id using the AUTO\_INCREMENT feature. Include columns for product\_name and price.

### C: NOT NULL

Create a table students that includes a student\_id (INT) and a student\_name (VARCHAR). Ensure that the student\_name column cannot have a NULL value by applying the NOT NULL constraint.

### D: UNIQUE

Create a table users that has a user\_id (INT) and email (VARCHAR). Ensure that no two users can have

the same email address by applying the UNIQUE constraint to the email column.

## E: FOREIGN KEY

Create two tables: one called departments and the other called employees. Each department has a department\_id as its primary key. In the employees table, include a column called department\_id as a foreign key that references the departments table. Ensure that every employee is linked to a department.

## Solution

[Lösung](#)

## Vocabulary

English	German
...	...
...	...



Volkan Demir

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/modul/m290/learningunits/lu05/aufgaben/03?rev=1727684810>

Last update: **2024/09/30 10:26**

