

Courses B0B36DBS, A7B36DBS: **Database Systems**

Practical Class 04:

SQL: DDL

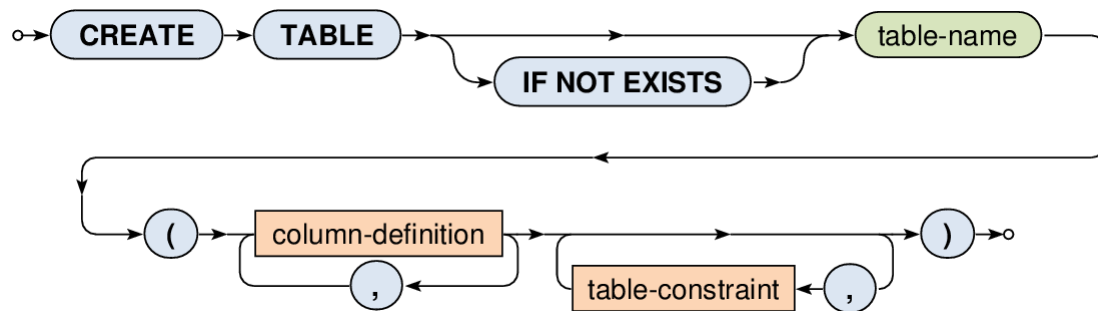
Martin Svoboda

14. 3. 2017

Faculty of Electrical Engineering, Czech Technical University in Prague

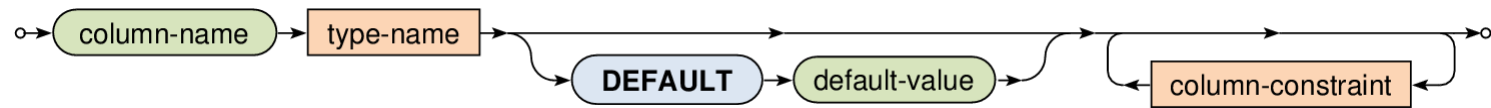
Tables

- **CREATE TABLE**
 - Creates schema for a new table



Columns

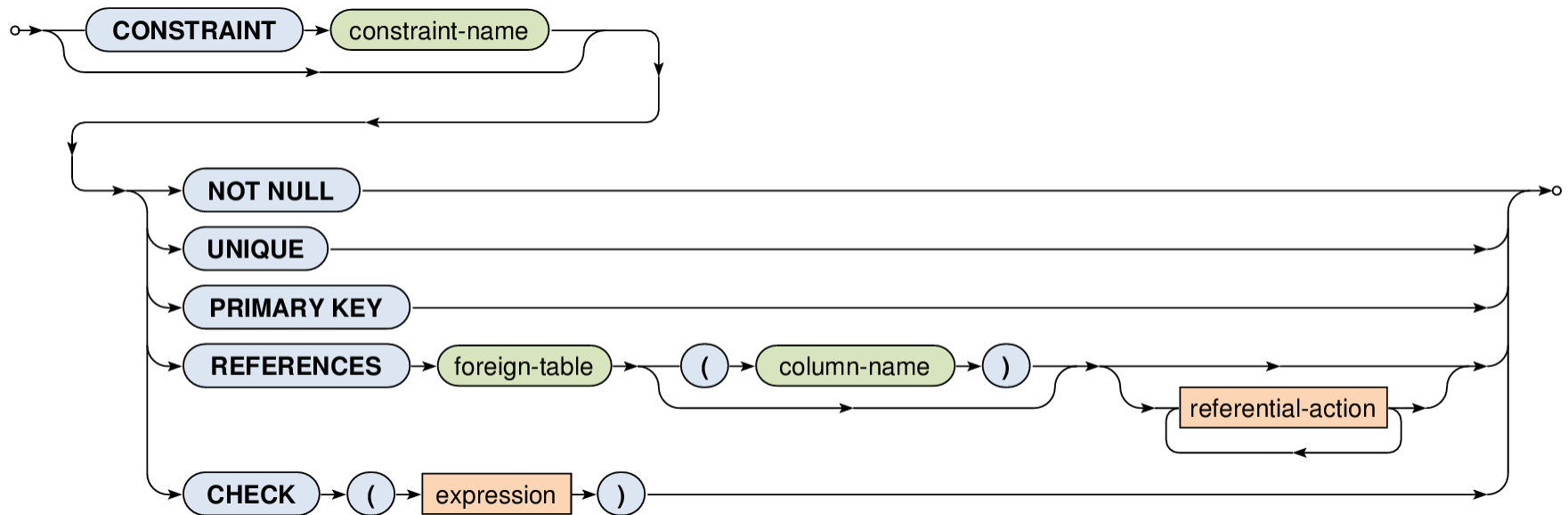
- **Column definition**



- **Data types** for (not only) columns
 - Precise numeric: **INT**, SMALLINT, BIGINT, **DECIMAL**
 - Approximate numeric: FLOAT, DOUBLE PRECISION
 - Logical: BOOLEAN
 - Character: **CHAR**, **VARCHAR**
 - Temporal: **DATE**, **TIME**, **DATETIME**
 - ...

Constraints

- **Column-level** integrity constraints

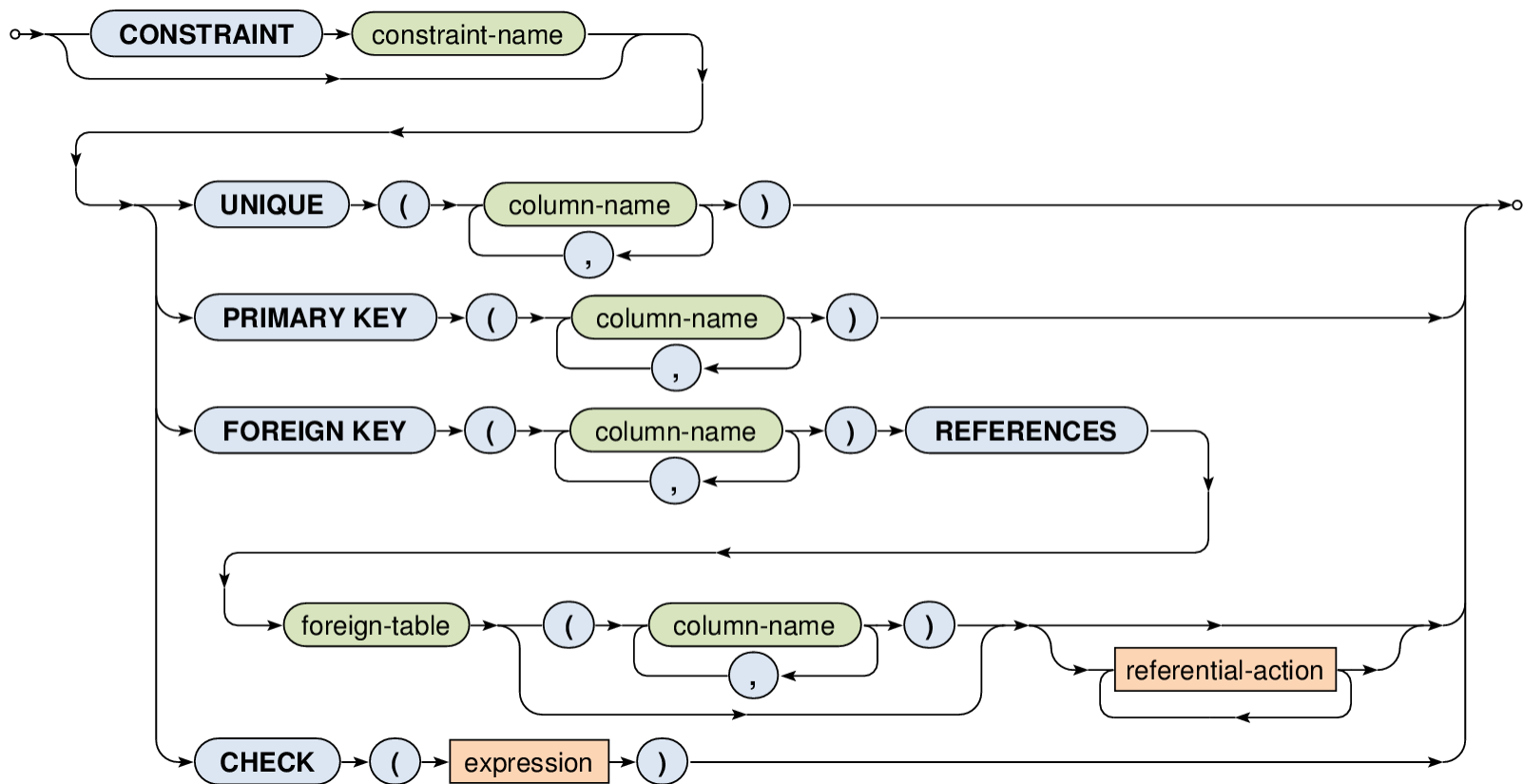


Exercise 1

- Create a schema for the following table:
 - **Library(Name, Street, City, PostCode)**
 - Choose appropriate data types
 - Define PRIMARY KEY and NOT NULL constraints
 - Use column-level constraints only

Constraints

- Table-level integrity constraints



Exercise 2

- Create a schema for the following table:
 - **User(Card, FirstName, LastName, Email, DateOfBirth)**
 - Card is a 16 digit long user card identification number
 - Date of birth is just optional
 - Describe all basic integrity constraints
 - Use table-level constraints only
 - Check email addresses for correctness
 - I.e. verify they correspond to a general pattern of e-mail addresses (using predicate LIKE)

Exercise 3

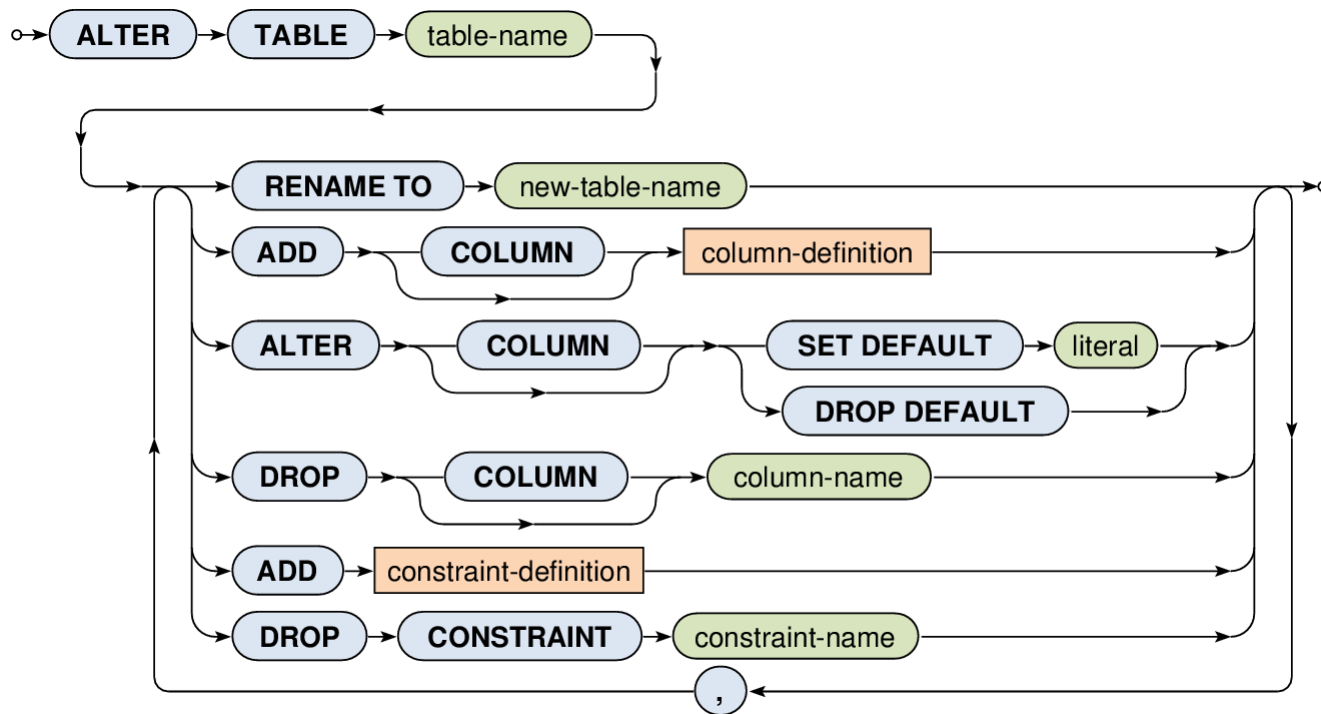
- Create a schema for the following table:
 - **Phone(User, Number)**
User $\subseteq$ User.Card
 - Phone numbers are always 9 digits long
 - Describe referential integrity
 - Use a column-level constraint for this purpose

Exercise 4

- Create schemata for the following tables:
 - **Title(IdTitle, ISBN, Title)**
 - IdTitle is an artificially generated integer identifier
 - ISBN identifiers are at most 17 characters long
 - Represent both the relational keys
 - **Author(IdAuthor, Name, YearOfBirth, YearOfDeath)**
 - Both years of birth and death are optional
 - Check mutual consistency of their values
 - **Authorship(Title, Author)**
Title $\subseteq$ Title.IdTitle, Author $\subseteq$ Author.IdAuthor
 - Use table-level constraints for the referential integrity

Tables

- **ALTER TABLE**
 - Modifies schema of a given table



Exercise 5

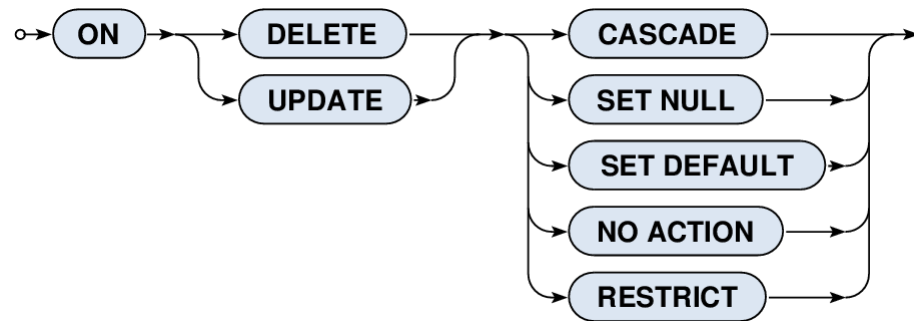
- Alter schema of a table for libraries:
 - Add IdLibrary as a new identifier of libraries

Exercise 6

- Create a schema for the following table:
 - **Book(Library, Signature, Title, DateOfAcquisition)**
Library $\subseteq$ Library.IdLibrary
Title $\subseteq$ Title.IdTitle

Constraints

- Referential actions



Exercise 7

- Create a schema for the following table:
 - **Loan(User, Library, Signature, TimeBorrowed, IdLoan, DateReturned)**
User $\subseteq$ User.Card
(Library, Signature) $\subseteq$ Book.(Library, Signature)
 - Return date is filled in when successfully returned
 - Add suitable referential actions
 - When a book / user is...
 - ... updated then the corresponding loans will be updated too
 - ... removed then the corresponding loans will be preserved

PostgreSQL

- **PostgreSQL**
 - Open-source ORDBMS
 - <https://www.postgresql.org/>
 - **pgAdmin4**
 - Administration and development platform
 - <https://www.pgadmin.org/>