

# SQL: Data Definition (3)

## Structured Query Language (SQL)

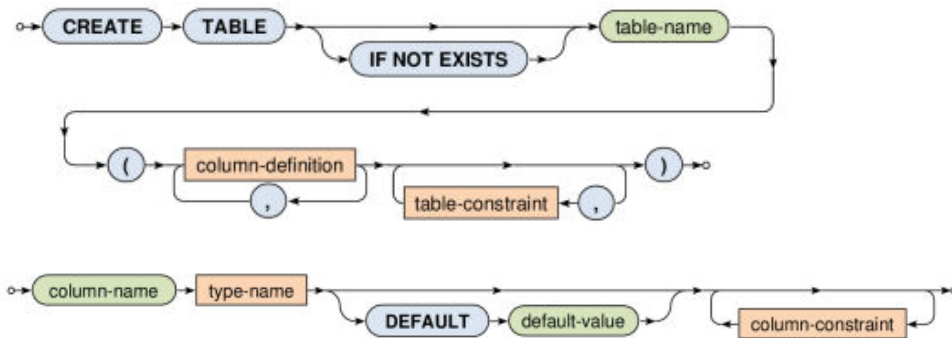
- standardizovaný jazyk pro přístup k relačním DB
- Data Definition (DDL) – tvorba tabulkových schémat a integritních omezení
- Data Manipulation (DML) – dotazování, inserty, mazání, updaty,...
- správa transakcí, administrace DB apod.
- je zpětně kompatibilní
- různé implementace ale následují různé úrovně standardů
- syntaktické diagramy
  - o grafická reprezentace bezkontextové gramatiky
  - o grafický popis jazyků (SQL,...)



## SQL: Schema Definition

### Tvorba tabulky

- konstrukce tabulkového schématu (a prázdné tabulky)
- jméno tabulky
- definice sloupců tabulky
  - o jméno sloupce
  - o typ dat
  - o defaultní hodnota (přidání nového řádku bez specifikovaných hodnot)
  - o definice sloupcových integritních omezení



## Example

- Simple table without integrity constraints

```
CREATE TABLE Product (  
  Id INTEGER,  
  Name VARCHAR(128),  
  Price DECIMAL(6,2),  
  Produced DATE,  
  Available BOOLEAN DEFAULT TRUE,  
  Weight FLOAT  
);
```

## Typy dat

### - Precise numeric types

- o INTEGER, INT, SMALLINT, BIGINT
- o DECIMAL(precision, scale)
  - precision = počet všech číslic (včetně desetinných)
  - scale = počet desetinných číslic

### - Approximate numeric types

- o FLOAT, REAL, DOUBLE PRECISION – reálná čísla

### - Logical values

- o BOOLEAN

### - Character strings

- o CHAR(délka), CHARACTER(délka) – řetězce pevné délky
  - kratší řetězce přidá doprava mezery
- o VARCHAR(délka), CHARACTER VARYING(délka) – řetězce proměnné délky

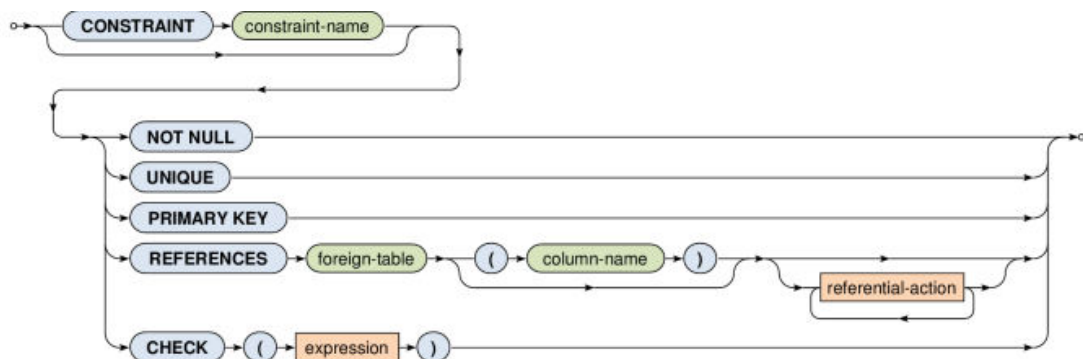
### - Temporal types

- o DATE, TIME, TIMESTAMP

```
CREATE TABLE Product (  
  Id INTEGER,  
  Name VARCHAR(128),  
  Price DECIMAL(6,2),  
  Produced DATE,  
  Available BOOLEAN DEFAULT TRUE,  
  Weight FLOAT  
);
```

## Integritní omezení

Sloupcová integritní omezení – umožňují omezit domény povolených hodnot



- **NOT NULL** – hodnoty nesmí být NULL
- **UNIQUE** – hodnoty musí být unikátní
- **PRIMARY KEY** – ekvivalent k NOT NULL + UNIQUE, může být jen jeden na tabulku
- **FOREIGN KEY** – referenční integrita
  - o hodnoty z odkazující tabulky musí existovat i v odkazované tabulce
  - o NULL hodnoty jsou ignorovány
  - o mohou být odkazovány jen unikátní / primární klíče
- **CHECK** – podmínka, která musí být splněna (lze testovat jen hodnoty v dané řádce)

```
CREATE TABLE Producer (  
  Id INTEGER PRIMARY KEY,  
  Name VARCHAR(128),  
  Country VARCHAR(64)  
);
```

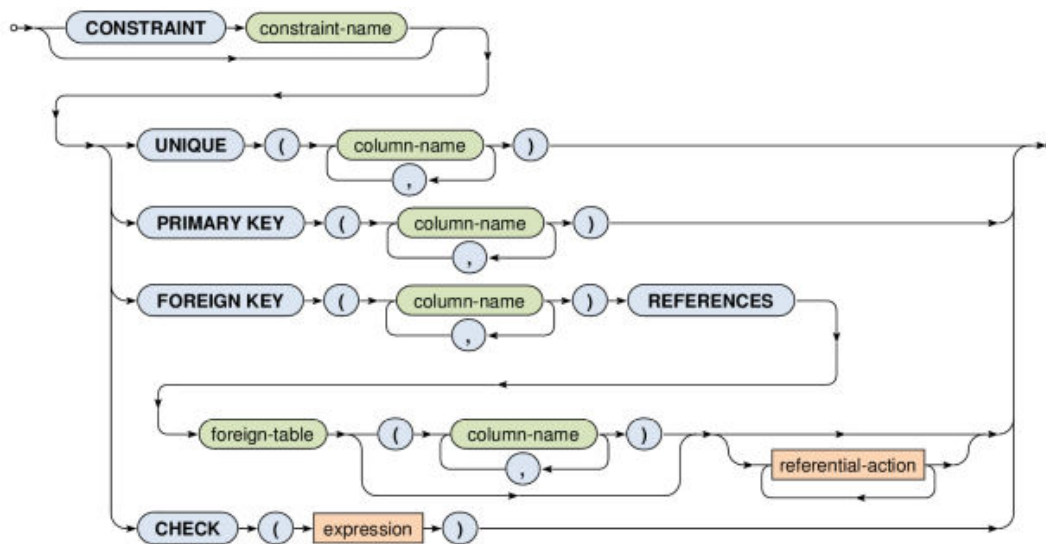
```
CREATE TABLE Product (
  Id INTEGER CONSTRAINT IC_Product_PK PRIMARY KEY,
  Name VARCHAR(128) UNIQUE,
  Price DECIMAL(6,2) CONSTRAINT IC_Product_Price NOT NULL,
  Produced DATE CHECK (Produced >= '2015-01-01'),
  Available BOOLEAN DEFAULT TRUE NOT NULL,
  Weight FLOAT,
  Producer INTEGER REFERENCES Producer (Id)
);
```

## ■ Referential integrity within a single table

```
CREATE TABLE Employee (
  Id INTEGER PRIMARY KEY,
  Name VARCHAR(128),
  Boss INTEGER REFERENCES Employee (Id)
);
```

Tabulková integritní omezení - pro vícero sloupců

- **UNIQUE**
- **PRIMARY KEY**
- **FOREIGN KEY** – tuply s alespoň jednou NULL budou ignorovány
- **CHECK** – lze testovat celé tabulky

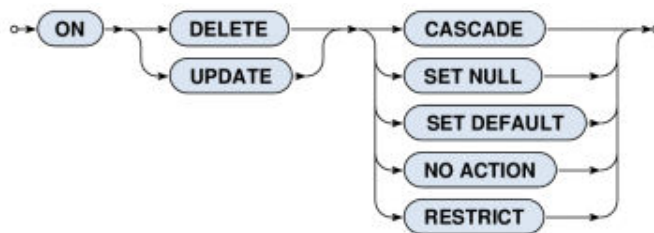


```
CREATE TABLE Producer (
  Name VARCHAR(128),
  Country VARCHAR(3),
  CONSTRAINT IC_Producer_PK PRIMARY KEY (Name, Country)
);
```

```
CREATE TABLE Product (
    Id INTEGER PRIMARY KEY,
    ...
    ProducerName VARCHAR(128),
    ProducerCountry VARCHAR(3),
    CONSTRAINT IC_Product_Producer_FK
        FOREIGN KEY (ProducerName, ProducerCountry)
        REFERENCES Producer (Name, Country)
);
```

## Referenční integrita

- pokud by operace na odkazované tabulce způsobila narušení cizího klíče v odkazující tabulce
  - o tj. hodnota cizího klíče alespoň jedné řádky v odkazující tabulce by se stala neplatnou
- ... tak:
  - o je operace zablokována a je vygenerována chybová hláška
  - o ale je-li definována referenční akce, je spuštěna



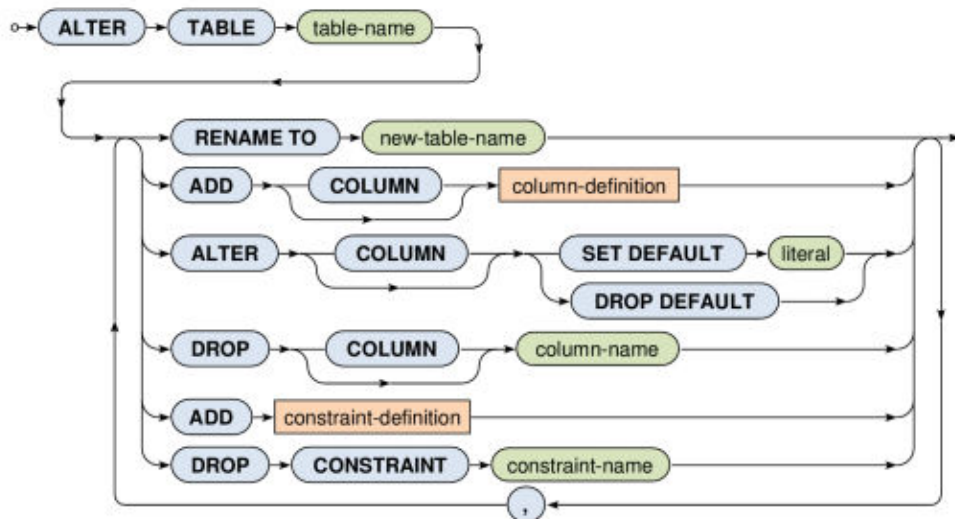
- **spouštěcí akce** (na odkazované tabulce):
  - o **ON UPDATE, ON DELETE**
- **referenční akce**
  - o **CASCADE** – řádka s odkazující hodnotou je updatována/smazána též
  - o **SET NULL**
  - o **SET DEFAULT**
  - o **NO ACTION** – jako by žádná referenční akce nebyla definována
  - o **RESTRICT** – žádná akce, ale kontrola ještě před pokusem o vykonání operace

```
CREATE TABLE Producer (
    Id INTEGER PRIMARY KEY,
    Name VARCHAR(128),
    Country VARCHAR(64)
);
```

```
CREATE TABLE Product (
    Id INTEGER PRIMARY KEY,
    ...
    Producer INTEGER
    REFERENCES Producer (Id) ON DELETE CASCADE
);
```

## ALTER TABLE

- Addition/change/removal of table columns/IC



## DROP TABLE

- Complementary to the table creation
  - I.e. table definition as well as table content are deleted

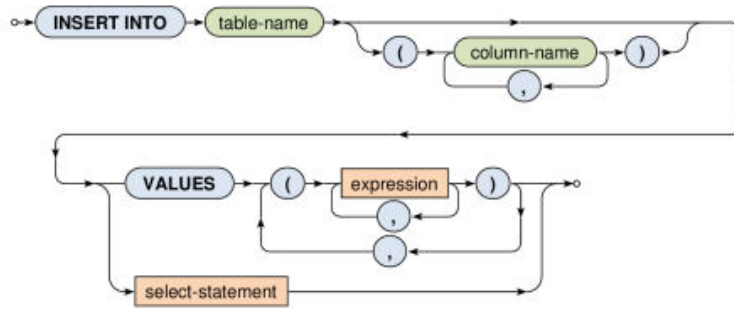


## SQL: Data Manipulation

- **modifikace dat:**
  - INSERT INTO
  - DELETE FROM
  - UPDATE
- **dotazování dat:**
  - SELECT

### Data insertion

- **INSERT INTO**
- vložení nových řádek do tabulky...
- ... explicitní enumerací / z výsledku výběru
- defaultní hodnoty pro vynechané sloupce



```

CREATE TABLE Product (
    Id INTEGER PRIMARY KEY,
    Name VARCHAR(128) UNIQUE,
    Price DECIMAL(6,2) NOT NULL,
    Produced DATE,
    Available BOOLEAN DEFAULT TRUE,
    Weight FLOAT,
    Producer INTEGER
);
  
```

### INSERT INTO Product

```
VALUES (0, 'Chair1', 2000, '2015-05-06', TRUE, 3.5, 11);
```

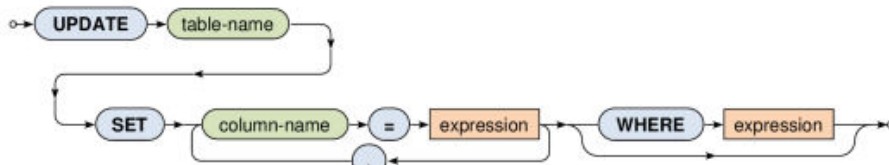
### INSERT INTO Product

```

(Id, Name, Price, Produced, Weight, Producer)
VALUES (1, 'Chair2', 1500, '2015-05-06', 4.5, 11);
  
```

## Data updates

- **UPDATE**
- modifikace existujících řádek v tabulce
  - o bere v úvahu jen řádky splňující danou podmínku
- nově přiřazené hodnoty mohou být:
  - o NULL, literal, hodnota daná výrazem, výsledek skalárního poddotazu,...



```

CREATE TABLE Product (
    Id INTEGER PRIMARY KEY,
    Name VARCHAR(128) UNIQUE,
    Price DECIMAL(6,2) NOT NULL,
    Produced DATE,
    Available BOOLEAN DEFAULT TRUE,
    Weight FLOAT,
    Producer INTEGER
);
  
```

### UPDATE Product

```

SET Name = 'Notebook'
WHERE (Name = 'Laptop');
  
```

### UPDATE Product

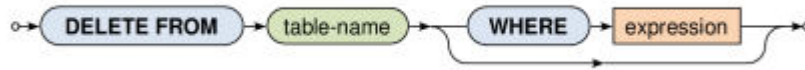
```

SET Price = Price * 0.9
WHERE (Produced < '2015-01-01');
  
```



## Data deletion

- DELETE FROM
- smazání existujících řádek z tabulky
- jen řádky splňující danou podmínku



```
CREATE TABLE Product (  
    Id INTEGER PRIMARY KEY,  
    Name VARCHAR(128) UNIQUE,  
    Price DECIMAL(6,2) NOT NULL,  
    Produced DATE,  
    Available BOOLEAN DEFAULT TRUE,  
    Weight FLOAT,  
    Producer INTEGER  
);
```

```
DELETE FROM Product  
    WHERE (Price > 2000);
```

```
DELETE FROM Product;
```