DDL-Ausdrücke und ihre Erklärungen

1. Tabelle: Studenten

```
CREATE TABLE Studenten (
    MatrNr INTEGER PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Semester INTEGER CHECK (Semester > 0)
);
```

Erklärung:

- **MatrNr**: Primärschlüssel zur eindeutigen Identifizierung jedes Studenten.
- **Name**: Name des Studenten, darf nicht NULL sein.
- **Semester**: Semester des Studenten, muss größer als 0 sein.

---

2. Tabelle: Professoren

```
CREATE TABLE Professoren (
    PersNr INTEGER PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Raum VARCHAR(10),
    VVorl INTEGER NOT NULL
    -- FOREIGN KEY (VVorl) REFERENCES Vorlesungen(VorlNr) -- jeder Professor muss m
);
```

Erklärung:

- **PersNr**: Primärschlüssel zur eindeutigen Identifizierung jedes Professors.
- **Name**: Name des Professors, darf nicht NULL sein.
- **Raum**: Raum des Professors.
- **VVorl**: Vorlesungsnummer, für die der Professor verantwortlich ist. Dieser Wert darf nicht NULL sein und verweist auf die Tabelle `Vorlesungen`.

INFO:

- Fremdschlüssel wird mit `ALTER TABLE` später hinzugefügt, da wir Tabelle Vorlesungen noch

nicht erstellt haben!

- **siehe letzter Eintrag in dieser Datei!**

---

3. Tabelle: Vorlesungen

```
CREATE TABLE Vorlesungen (
    VorlNr INTEGER PRIMARY KEY,
    Titel VARCHAR(100) NOT NULL,
    SWS INTEGER CHECK (SWS > 0),
    gelesenVon INTEGER NOT NULL
    -- FOREIGN KEY (gelesenVon) REFERENCES Professoren(PersNr) -- jede Vorlesung mus
);
```

Erklärung:

- **VorlNr**: Primärschlüssel zur eindeutigen Identifizierung jeder Vorlesung.
- **Titel**: Titel der Vorlesung, darf nicht NULL sein.
- **SWS**: Semesterwochenstunden der Vorlesung, müssen größer als 0 sein.
- **gelesenVon**: Personalnummer des Professors, der die Vorlesung liest. Dieser Wert darf nicht NULL sein und verweist auf die Tabelle `Professoren`.

INFO:

- Fremdschlüssel wird mit `ALTER TABLE` später hinzugefügt, um eine klare Übersicht der gegenseitigen Abhängigkeit der Tabellen `Vorlesungen` und `Professoren` zu gewährleisten! Da die Tabelle `Professoren` bereits erzeugt wurde kann der Fremdschlüssel auch direkt bei Erzeugung der `Vorlesungen` Tabelle mitgegeben werden!
- **siehe letzter Eintrag in dieser Datei!**

---

4. Tabelle: Assistenten

```
CREATE TABLE Assistenten (
    PersNr INTEGER PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Fachgebiet VARCHAR(100),
    Boss INTEGER NOT NULL,
    FOREIGN KEY (Boss) REFERENCES Professoren(PersNr)
);
```

Erklärung:

- **PersNr**: Primärschlüssel zur eindeutigen Identifizierung jedes Assistenten.
- **Name**: Name des Assistenten, darf nicht NULL sein.
- **Fachgebiet**: Fachgebiet des Assistenten.
- **Boss**: Personalnummer des Professors, dem der Assistent unterstellt ist. Dieser Wert darf nicht NULL sein und verweist auf die Tabelle `Professoren`.

---

5. Tabelle: hoeren

```
CREATE TABLE hoeren (
    MatrNr INTEGER,
    VorlNr INTEGER,
    PRIMARY KEY (MatrNr, VorlNr),
    FOREIGN KEY (MatrNr) REFERENCES Studenten(MatrNr),
    FOREIGN KEY (VorlNr) REFERENCES Vorlesungen(VorlNr)
);
```

Erklärung:

- **MatrNr**: Matrikelnummer des Studenten, verweist auf die Tabelle `Studenten`.
- **VorlNr**: Vorlesungsnummer, verweist auf die Tabelle `Vorlesungen`.
- **PRIMARY KEY (MatrNr, VorlNr)**: Zusammengesetzter Primärschlüssel aus MatrNr und VorlNr, stellt sicher, dass jede Kombination aus MatrNr und VorlNr einzigartig ist.

---

6. Tabelle: voraussetzen

```
CREATE TABLE voraussetzen (
    Vorgaenger INTEGER,
    Nachfolger INTEGER,
    PRIMARY KEY (Vorgaenger, Nachfolger),
    FOREIGN KEY (Vorgaenger) REFERENCES Vorlesungen(VorlNr),
    FOREIGN KEY (Nachfolger) REFERENCES Vorlesungen(VorlNr)
);
```

Erklärung:

- **Vorgaenger**: Vorlesungsnummer der vorausgehenden Vorlesung, verweist auf die Tabelle

`Vorlesungen`.

- **Nachfolger**: Vorlesungsnummer der nachfolgenden Vorlesung, verweist auf die Tabelle `Vorlesungen`.
- **PRIMARY KEY (Vorgaenger, Nachfolger)**: Zusammengesetzter Primärschlüssel aus Vorgaenger und Nachfolger, stellt sicher, dass jede Kombination aus Vorgaenger und Nachfolger einzigartig ist.

---

7. Tabelle: pruefen

```
CREATE TABLE pruefen (
    MatrNr INTEGER,
    VorlNr INTEGER,
    PersNr INTEGER,
    Note NUMERIC(2,1) CHECK (Note BETWEEN 1.0 AND 5.0),
    PRIMARY KEY (MatrNr, VorlNr),
    FOREIGN KEY (MatrNr) REFERENCES Studenten(MatrNr),
    FOREIGN KEY (VorlNr) REFERENCES Vorlesungen(VorlNr),
    FOREIGN KEY (PersNr) REFERENCES Professoren(PersNr)
);
```

Erklärung:

- **MatrNr**: Matrikelnummer des Studenten, verweist auf die Tabelle `Studenten`.
- **VorlNr**: Vorlesungsnummer, verweist auf die Tabelle `Vorlesungen`.
- **PersNr**: Personalnummer des Professors, der die Prüfung abnimmt, verweist auf die Tabelle `Professoren`.
- **Note**: Note der Prüfung, muss zwischen 1.0 und 5.0 liegen.
- **PRIMARY KEY (MatrNr, VorlNr)**: Zusammengesetzter Primärschlüssel aus MatrNr und VorlNr, stellt sicher, dass jede Kombination aus MatrNr und VorlNr einzigartig ist.

---

8. Fremdschlüsseleinschränkungen

```
ALTER TABLE Professoren
ADD CONSTRAINT fk_professoren_vv
FOREIGN KEY (VVorl) REFERENCES Vorlesungen(VorlNr);

ALTER TABLE Vorlesungen
ADD CONSTRAINT vorlesungen_gelesenvon_fkey
FOREIGN KEY (gelesenVon) REFERENCES Professoren(PersNr);
```

Erklärung:

- **fk_professoren_vv**: Fügt eine Fremdschlüsseleinschränkung hinzu, die sicherstellt, dass die Vorlesungsnummer `VVorl` in der Tabelle `Professoren` auf eine existierende Vorlesungsnummer in der Tabelle `Vorlesungen` verweist.
- **vorlesungen_gelesenvon_fkey**: Fügt eine Fremdschlüsseleinschränkung hinzu, die sicherstellt, dass die Personalnummer `gelesenVon` in der Tabelle `Vorlesungen` auf eine existierende Personalnummer in der Tabelle `Professoren` verweist.