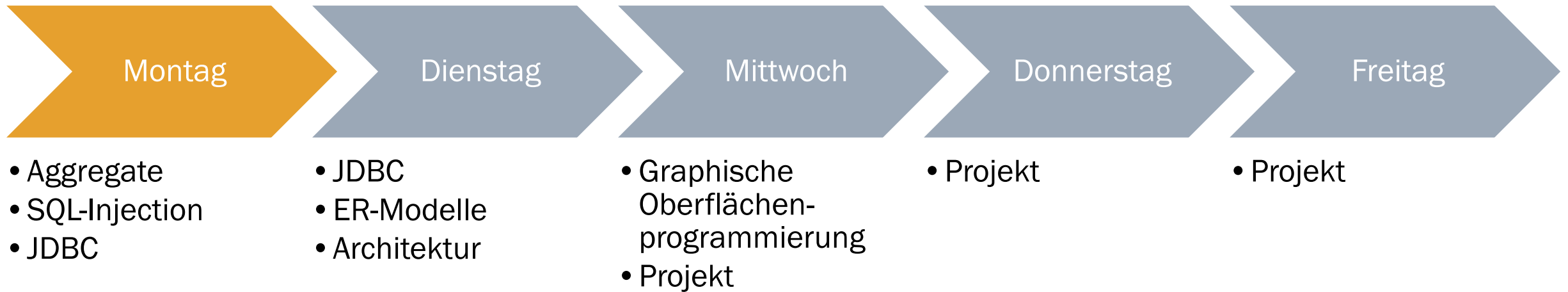


Einführung in Datenbanken

März 2025

Plan für die Woche



Plan für heute

- Was sind Aggregatfunktionen?
- Wie schreibt man Aggregatfunktionen?
- GROUP BY , HAVING
- Übungen

Aggregatfunktionen

Aggregatfunktionen

- mittels Aggregatfunktionen können Berechnungen an einem Set an Werten oder einem Wert durchgeführt werden
- mögliche Aggregatfunktionen:
 - MIN(column_name) – gibt den kleinsten Wert der Spalte aus
 - MAX(column_name) – gibt den größten Wert der Spalte aus
 - COUNT(column_name) – gibt die Anzahl an Datensätzen in einer Spalte aus
 - AVG(column_name) – gibt den Durchschnitt der numerischen Spalte aus
 - SUM(column_name) – gibt die Summe der numerischen Spalte aus

MIN() / MAX()



Aliasing
praktisch!

- wird genutzt, um den minimalen/maximalen Wert einer Spalte zu bestimmen
- Syntax:

```
SELECT MIN(column_name) AS minimum  
FROM table_name  
WHERE condition;
```

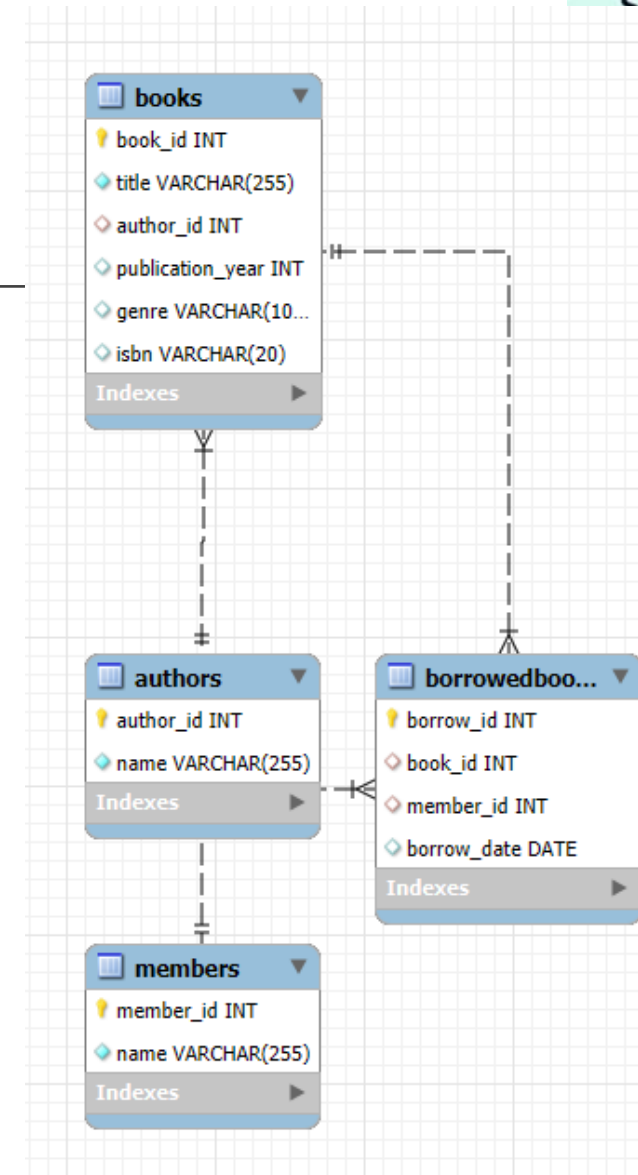
```
SELECT MAX(column_name) AS maximum  
FROM table_name  
WHERE condition;
```

DEMO MIN / MAX

Aufgaben

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

- Finde das älteste und das neueste Buch in der Bibliothek
- Finde das frühste und das späteste Ausleihdatum



COUNT()

- gibt die Anzahl an Datensätzen in einer Spalte zurück, die eine Kondition erfüllen muss
- Syntax:

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT COUNT(*)  
FROM table_name  
WHERE condition;
```

DISTINCT

- wird verwendet, um doppelte Werte aus dem Ergebnis einer Abfrage zu entfernen
- mit Aggregatfunktionen kombinierbar
- Syntax:

```
SELECT DISTINCT column_name  
FROM table_name  
WHERE condition;
```

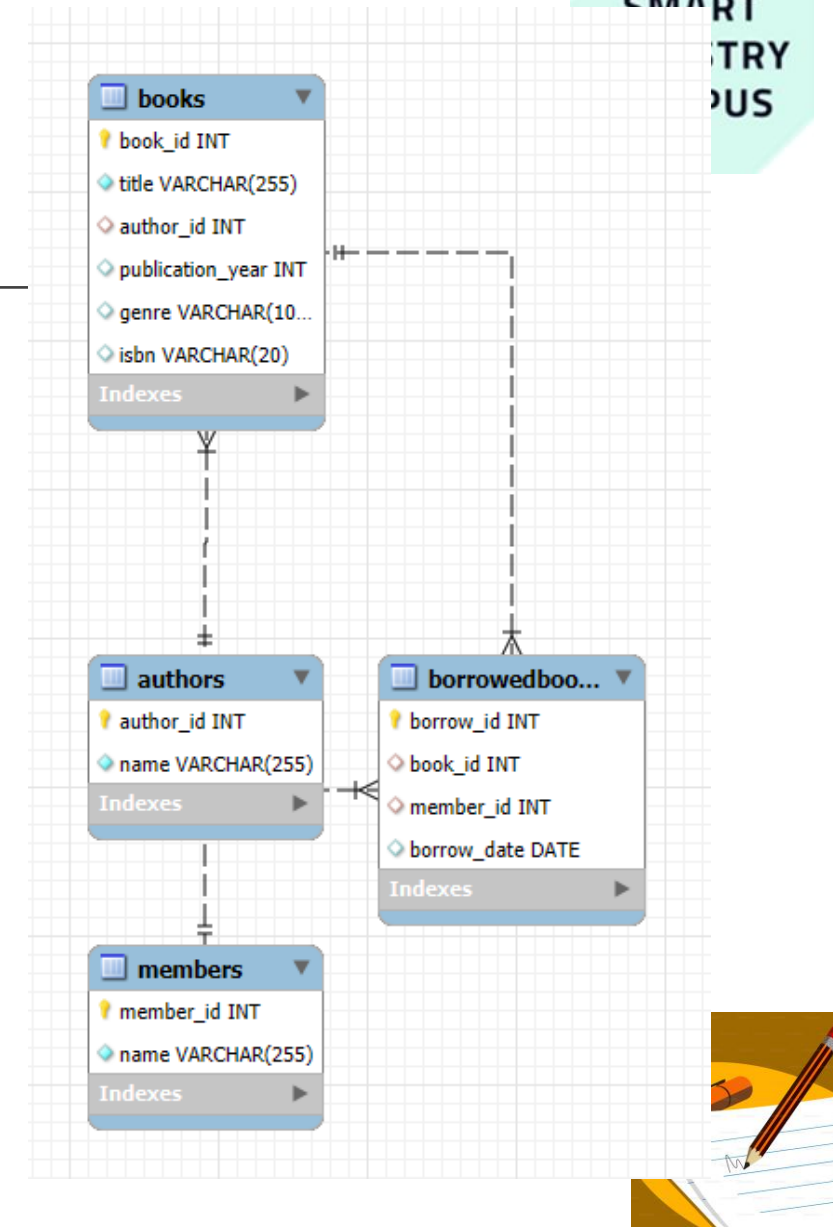
```
SELECT COUNT(DISTINCT column_name )  
FROM table_name  
WHERE condition;
```

DEMO DISTINCT/ COUNT

Aufgaben

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

- Zähle die Anzahl alle Bücher in der Bibliothek
- Zähle die Anzahl der Mitglieder in einer Bibliothek
- Zähle die Anzahl der ausgeliehenen Bücher
- Zeige alle verschiedenen Genres, die in der Bibliothek vorkommt
- Liste alle einzigartigen Jahre auf, in denen Bücher veröffentlicht wurden
- Zähle an wie vielen verschiedenen Tagen Bücher ausgeliehen wurden
- Zähle, wie viele Bücher nach dem 1.März 2024 ausgeliehen wurden



SUM() / AVG()

- gibt die Summe oder Durchschnitt einer **numerischen** Spalte zurück
- Syntax:

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```

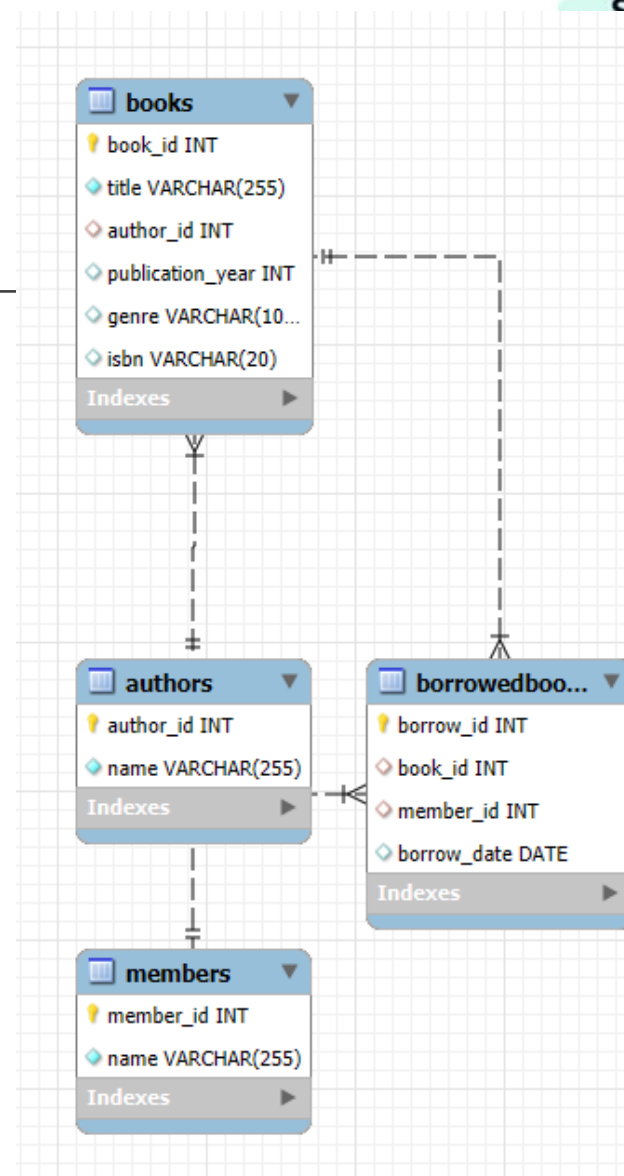
```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

DEMO SUM() / AVG()

Aufgaben

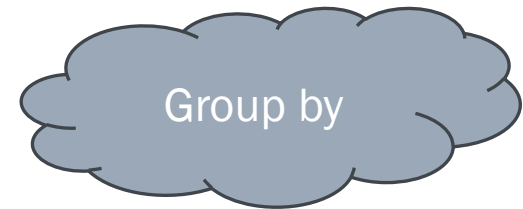
```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

- Berechne das durchschnittliche Erscheinungsjahr der Bücher



Anmerkung

- wenn Aggregatfunktionen verwendet werden, wird nur ein einzelnes Ergebnis zurückgegeben:
 - es werden mehrere Zeilen zu einem einzelnen Wert zusammengefasst
 - aus diesem Grund ist im SELECT-Statement nur eine Kombination aus Aggregatfunktionen möglich
- Eine Kombination aus Aggregatfunktionen und einzelnen Spalten ohne Gruppierung ist nicht zulässig
 - Die Datenbank weiß dann nicht, welche Zeile für diese Spalte genommen werden soll
- Beispiel:
 - Tabelle Patient -> man möchte das Durchschnittsalter und den Namen haben
 - Durchschnittsalter wird aus mehreren Zeilen berechnet -> einzelne Zeile als Resultat
 - Es gibt aber viele Namen -> welche Zeile soll ausgegeben werden?



GROUP BY

- Daten können in Gruppen zusammengefasst werden
- wird oft im Zusammenhang mit Aggregatfunktionen genutzt
- Beispiel:
 - Finde die Anzahl der Kunden **je Land**
 - Was ist das durchschnittliche Gehalt **je Abteilung?**
- Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
[ORDER BY column_name(s)];
```

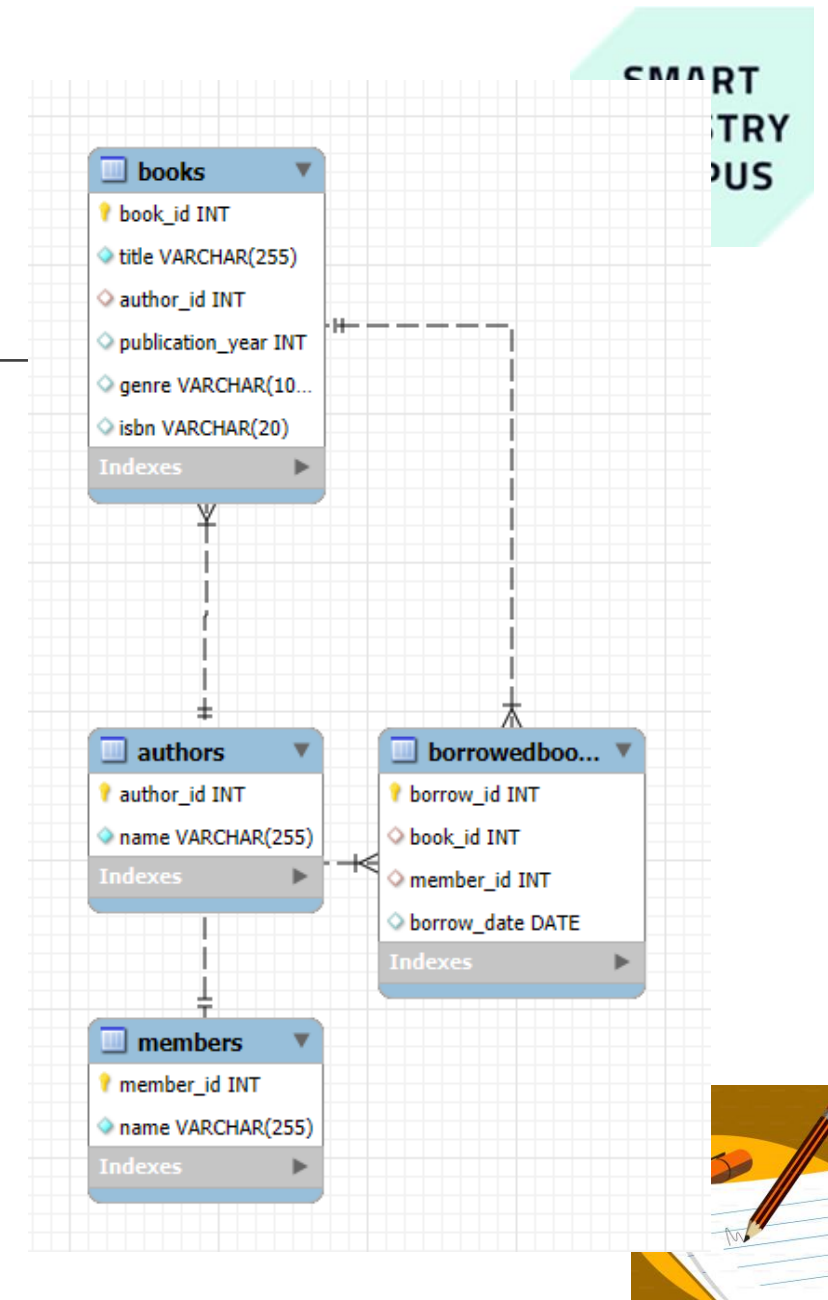
DEMO GROUP BY

Aufgaben

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
[ORDER BY column_name(s)];
```

- Zähle, wie viele Bücher jedes Genre in der Bibliothek hat

Fantasy	2
Dystopie	1
Allegorie	1
Klassiker	1
Coming-of-Age	1
Romanze	1
Drama	1



HAVING

- wurde eingeführt, da Aggregatfunktionen nicht mit WHERE genutzt werden können
- Beispiel:
 - Zeige alle Abteilungen, dessen durchschnittliches Gehalt über 50.000€ liegt
- Syntax:

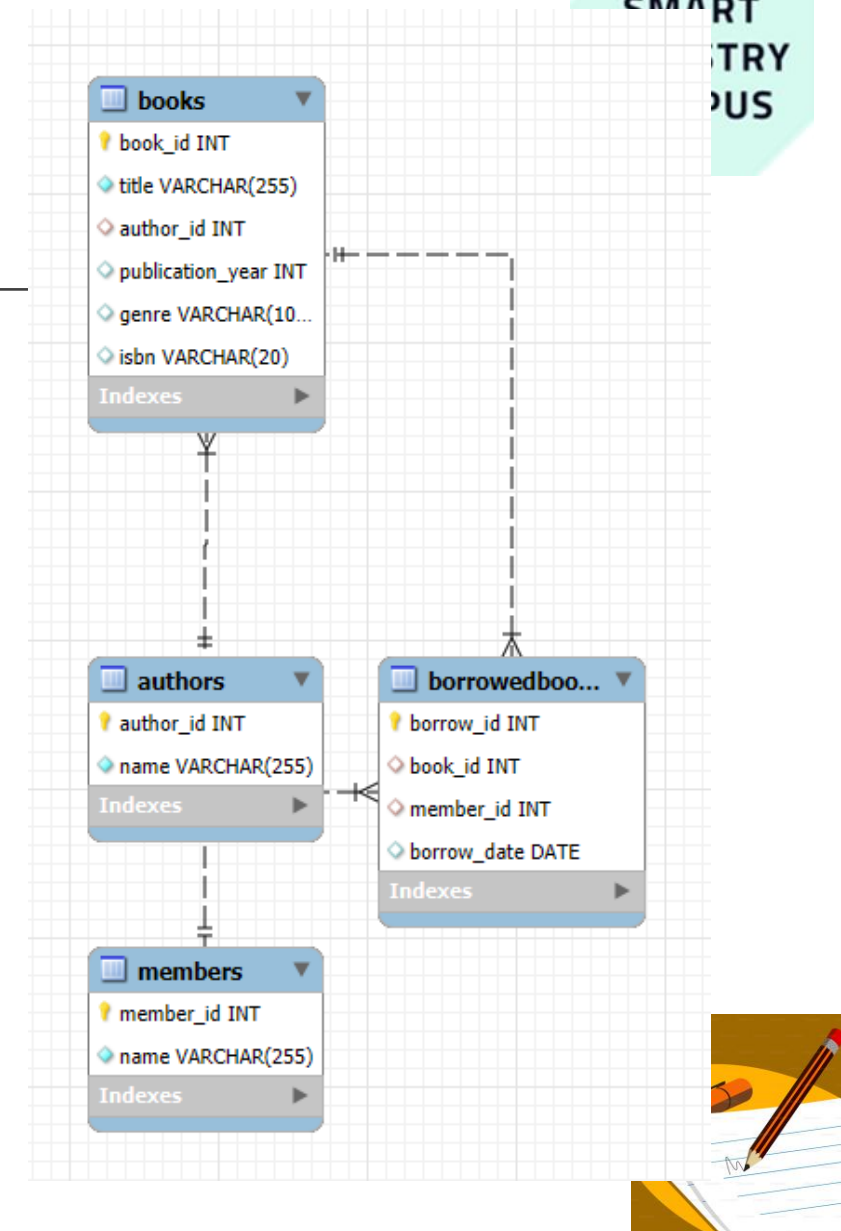
```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
[ORDER BY column_name(s)];
```

DEMO HAVING

Aufgaben

- Zeige alle Genres mit 2 Büchern

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
[ORDER BY column_name(s)];
```

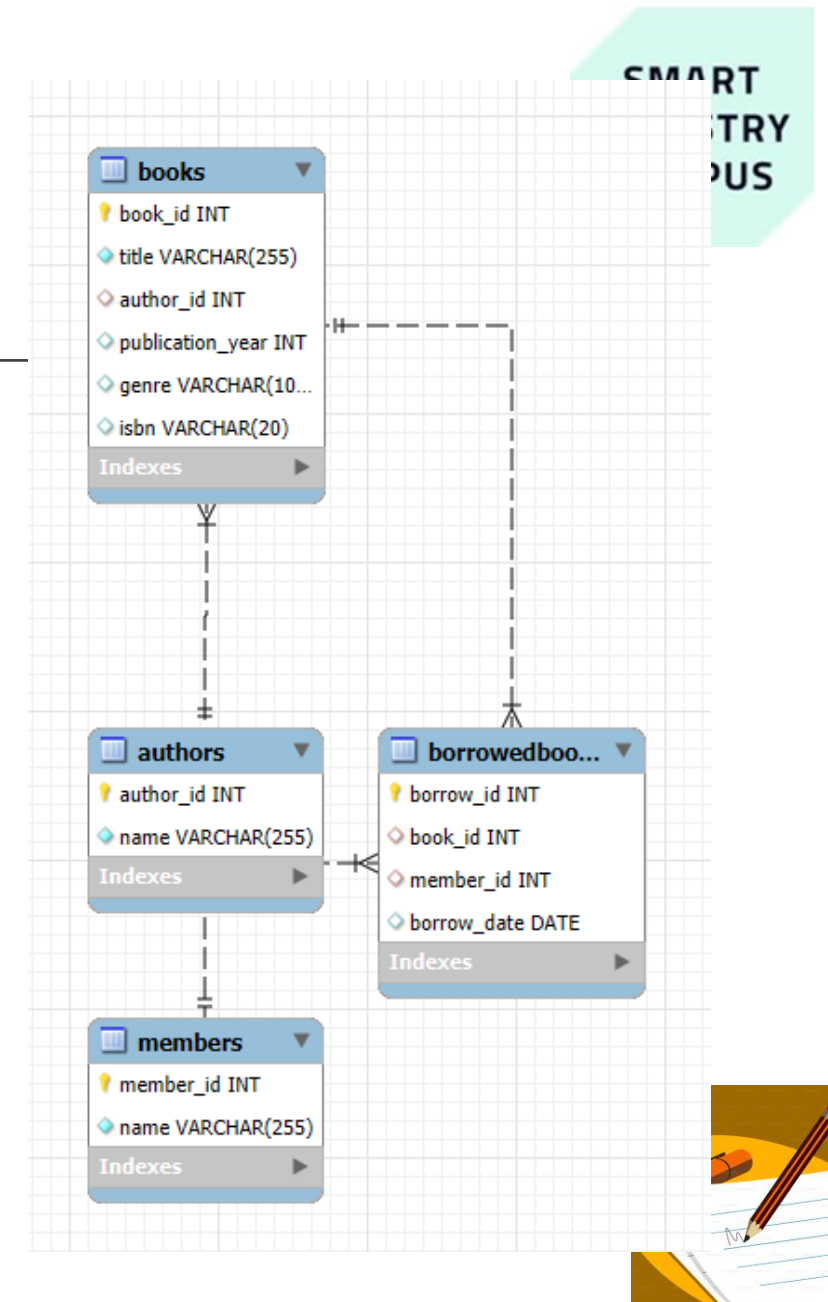


Aufgaben

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
[ORDER BY column_name(s)];
```

–Zeige Genres, in denen das älteste Buch vor 1950 veröffentlicht wurde:

	genre	oldest_book
▶	Fantasy	1937
	Dystopie	1949
	Allegorie	1945
	Klassiker	1925
	Romanze	1813



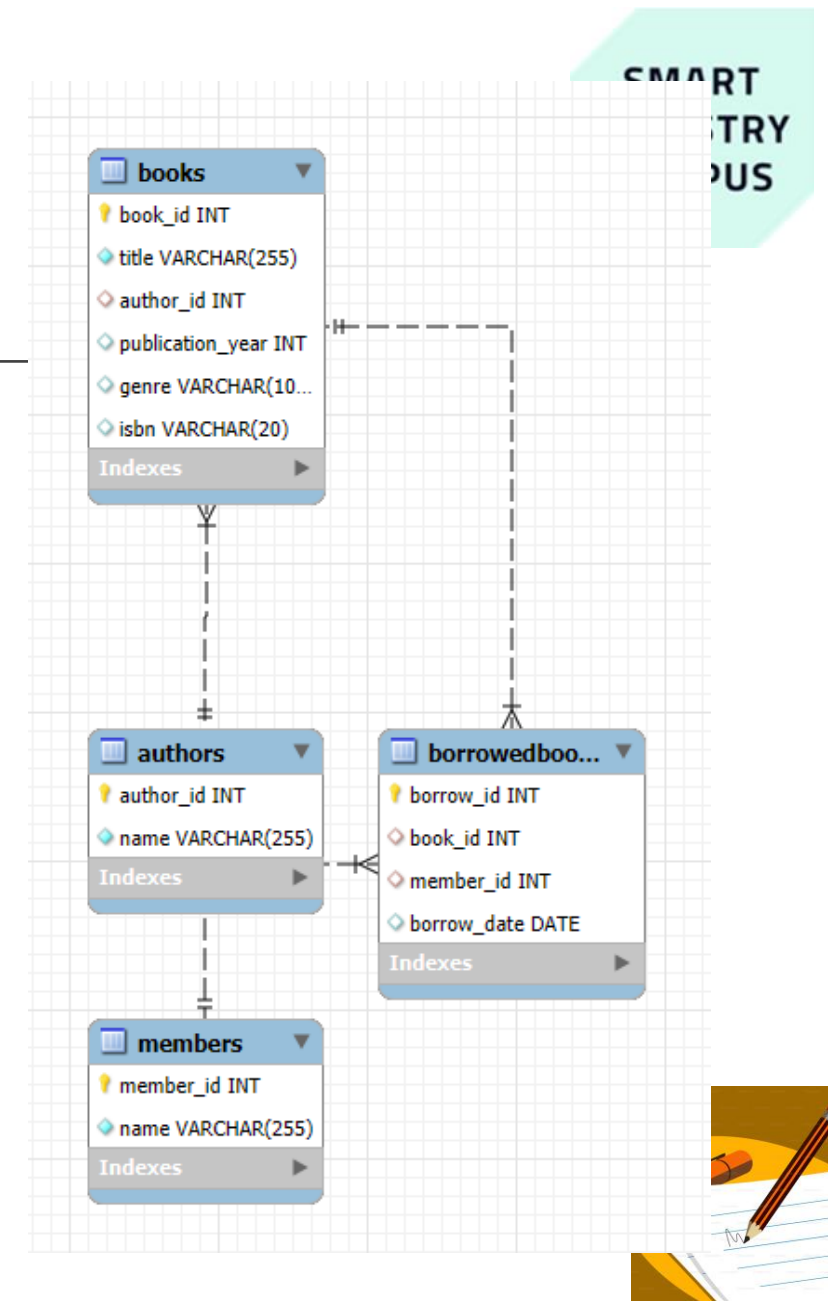
Nun alles zusammen

- Aggregatfunktionen, JOINS, WHERE, GROUP BY, ORDER BY und HAVING – alles kann zusammengesetzt in einer Query stehen

Aufgaben

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
[ORDER BY column_name(s)];
```

- Zeige, wie viele Bücher jedes Mitglied ausgeliehen hat
- Zeige alle Mitglieder an, die kein Buch ausgeliehen haben
- Zähle, wie viele Bücher jedes Mitglied ausgeliehen hat
- Zeige alle Bücher, die vor 1950 veröffentlicht wurden und von mehr als einem Mitglied ausgeliehen wurden
- Zeige alle Autoren, die mehr als ein Buch in der Bibliothek haben
- Zähle, wie viele Bücher von jedem Mitglied ausgeliehen wurden
- Zeige alle Bücher, die von mehr als einem Mitglied ausgeliehen wurden
- Zähle, wie viele Bücher jedes Mitglied nach dem 1. März 2024 ausgeliehen hat



Aufgaben

<https://sql-island.informatik.uni-kl.de/>

