

# Abfragen über mehrere Tabellen (JOINS)

Daten aus mehreren Tabellen kombinieren – so funktioniert's!

# Warum JOINS?

- Daten in relationalen Datenbanken sind **verteilt auf mehrere Tabellen**
- JOINS verbinden diese Tabellen logisch
- Ohne JOIN: viele wichtige Informationen fehlen

# Grundidee eines JOINS

```
SELECT * FROM Tabelle1
```

```
JOIN Tabelle2 ON Tabelle1.ID = Tabelle2.FremdID;
```

- JOIN verbindet Datensätze über eine **gemeinsame Spalte**
- Ergebnis: **eine neue Ergebnistabelle**

# Übersicht – JOIN-Arten

JOIN-Art	Beschreibung
INNER JOIN	Nur passende Datensätze
LEFT OUTER JOIN	Alle von links + passende von rechts
RIGHT OUTER JOIN	Alle von rechts + passende von links
FULL OUTER JOIN	Alle Datensätze, auch ohne Partner
NATURAL JOIN	Automatisch über gleiche Spaltennamen
SELF JOIN	Tabelle mit sich selbst verbinden

# INNER JOIN

- Zeigt **nur** Datensätze mit Übereinstimmungen in beiden Tabellen

```
SELECT * FROM Kunden
```

```
INNER JOIN Bestellungen
```

```
ON Kunden.ID = Bestellungen.KundenID;
```

Ergebnis:

Name	Artikel
Max	Laptop
Lisa	Smartphone

KundenID	Name
1	Max
2	Lisa
3	Tom

BestellungID	KundenID	Artikel
101	1	Laptop
102	2	Smartphone

# LEFT OUTER JOIN

- Zeigt **alle** Datensätze aus der **linken Tabelle**
- Rechte Tabelle: nur, wenn passende Daten vorhanden

```
SELECT * FROM Kunden
```

```
LEFT JOIN Bestellungen
```

```
ON Kunden.ID = Bestellungen.KundenID;
```

Ergebnis:

Name	Artikel
Max	Laptop
Lisa	Smartphone
Tom	NULL

KundenID	Name
1	Max
2	Lisa
3	Tom

BestellungID	KundenID	Artikel
101	1	Laptop
102	2	Smartphone

# RIGHT OUTER JOIN

- Zeigt **alle** Datensätze aus der **rechten Tabelle**
- Linke Tabelle: nur mit passenden Daten

```
SELECT * FROM Kunden
```

```
RIGHT JOIN Bestellungen
```

```
ON Kunden.ID = Bestellungen.KundenID;
```

Ergebnis:

Name	Artikel
Max	Laptop
Lisa	Smartphone
NULL	Tablet

KundenID	Name
1	Max
2	Lisa
3	Tom

BestellungID	KundenID	Artikel
101	1	Laptop
102	2	Smartphone
103	5	Tablet

# FULL OUTER JOIN

- Zeigt **alle** Datensätze aus **beiden Tabellen**
- Wenn kein Partner vorhanden -> **NULL**

KundenID	Name
1	Max
2	Lisa
3	Tom

```
SELECT * FROM Kunden
```

```
FULL OUTER JOIN Bestellungen
```

```
ON Kunden.ID = Bestellungen.KundenID;
```

BestellungID	KundenID	Artikel
101	1	Laptop
102	2	Smartphone
103	5	Tablet

Name	Artikel
Max	Laptop
Lisa	Smartphone
Tom	NULL
NULL	Tablet



# NATURAL JOIN

- Automatische Verbindung über Spalten mit **gleichen Namen**

```
SELECT * FROM Kunden
```

```
NATURAL JOIN Bestellungen;
```

ID	Name
1	Max
2	Lisa

ID	Lieferstatus
1	Zugestellt
3	In Bearbeitung

**Achtung:** Nur sinnvoll, wenn Spaltennamen und -bedeutung identisch sind.

Ergebnis:

ID	Name	Lieferstatus
1	Max	Zugestellt

# SELF JOIN

ID	Name	VorgesetzterID
1	Max	NULL
2	Lisa	1
3	Tom	1
4	Anna	2

- Eine Tabelle wird mit sich selbst verknüpft
- Beispiel: Mitarbeiter und ihre Vorgesetzten

```
SELECT A.Name AS Mitarbeiter, B.Name AS Vorgesetzter  
FROM Mitarbeiter A  
JOIN Mitarbeiter B ON A.VorgesetzterID = B.ID;
```

Ergebnis:

Mitarbeiter	Vorgesetzter
Lisa	Max
Tom	Max
Anna	Lisa

# Zusammenfassung

<b>JOIN</b>	<b>Zeigt</b>
INNER	Nur gemeinsame Daten
LEFT	Alle links + passende rechts
RIGHT	Alle rechts + passende links
FULL	Alle Kombinationen
NATURAL	Automatische Verbindung
SELF	Tabelle mit sich selbst

# Videotipp

- [Joins in SQL. Einfach erklärt \(Left-Join, Right-Join, Cross-Join, Left-Anti-Join\)](#)