

Kapitel 8 - IHK-Aufgaben und Übersichten

- [8.1 ERM-Aufgaben lösen](#)
- [8.2 SQL-Aufgaben lösen](#)
- [8.3 JOIN-Übersicht](#)
- [8.4 Normalformen-Übersicht](#)
- [8.5 Häufige Verwechslungen](#)

8.1 ERM-Aufgaben lösen

Definition

Bei **ERM-Aufgaben** wird aus einer Beschreibung ein Datenmodell erstellt.

Dabei werden Entitäten, Attribute, Beziehungen und Kardinalitäten bestimmt.

Vorgehen

Schritt	Aufgabe
1	Entitäten erkennen
2	Attribute zuordnen
3	Primärschlüssel festlegen
4	Beziehungen bestimmen
5	Kardinalitäten eintragen
6	N:M-Beziehungen auflösen
7	Relationenmodell ableiten

Typische Entitäten

Entitäten sind häufig Substantive im Aufgabentext.

Beispiele:

- Kunde
- Bestellung
- Artikel
- Rechnung
- Mitarbeiter
- Abteilung

Typische Beziehungen

Beziehungen ergeben sich häufig aus Verben im Aufgabentext.

Beispiele:

- Kunde gibt Bestellung auf
- Bestellung enthält Artikel

- Mitarbeiter gehört zu Abteilung
- Rechnung bezieht sich auf Bestellung

Kardinalitäten

Aussage im Text	Typische Kardinalität
genau ein	1
mehrere	n
kann mehrere	n
gehört zu genau einem	1
enthält mehrere	n

Merksatz

Substantive werden oft zu Entitäten.
Verben zeigen häufig Beziehungen.

Prüfungshinweis

In IHK-Aufgaben wird häufig ein Text vorgegeben, aus dem ein ERM oder Relationenmodell erstellt werden muss.

Typische Fehler:

- Attribute als Entitäten darstellen.
- Beziehungen vergessen.
- Kardinalitäten falsch bestimmen.
- N:M-Beziehungen nicht auflösen.
- Primär- und Fremdschlüssel nicht kennzeichnen.

Quellen

- IT-Handbuch für Fachinformatiker

8.2 SQL-Aufgaben lösen

Definition

Bei **SQL-Aufgaben** wird aus einer Aufgabenstellung eine passende SQL-Abfrage erstellt.

Dabei müssen Tabellen, Spalten, Bedingungen und ggf. JOINS erkannt werden.

Vorgehen

Schritt	Aufgabe
1	Gesuchte Ausgabe erkennen
2	Benötigte Tabelle bestimmen
3	Benötigte Spalten auswählen
4	Bedingungen aus dem Text ableiten
5	Falls nötig JOIN verwenden
6	Falls nötig gruppieren oder sortieren
7	SQL-Reihenfolge prüfen

Typische Signalwörter

Formulierung	SQL-Bestandteil
„zeige“ / „liste“	SELECT
„aus Tabelle“	FROM
„nur“ / „bei denen“	WHERE
„sortiert nach“	ORDER BY
„Anzahl“	COUNT
„Summe“	SUM
„Durchschnitt“	AVG
„pro Kunde / je Ort“	GROUP BY
„mehr als / weniger als“ bei Gruppen	HAVING

Beispiel

Aufgabe:

Alle Kunden aus Berlin anzeigen.

```
SELECT *  
FROM kunde  
WHERE ort = 'Berlin';
```

Beispiel mit Sortierung

Aufgabe:

Alle Artikel nach Preis absteigend sortieren.

```
SELECT *  
FROM artikel  
ORDER BY preis DESC;
```

Merksatz

Erst verstehen, was ausgegeben werden soll, dann SQL schreiben.

Prüfungshinweis

In IHK-Aufgaben ist die richtige SQL-Reihenfolge besonders wichtig.

Typische Fehler:

- SELECT-Spalten falsch auswählen.
- WHERE-Bedingung vergessen.
- Textwerte ohne Anführungszeichen schreiben.
- ORDER BY an die falsche Stelle setzen.
- GROUP BY und HAVING verwechseln.
- JOIN vergessen, obwohl Daten aus mehreren Tabellen benötigt werden.

Quellen

- IT-Handbuch für Fachinformatiker

8.3 JOIN-Übersicht

Definition

Ein **JOIN** verbindet Datensätze aus mehreren Tabellen.

JOINS werden verwendet, wenn benötigte Daten auf mehrere Tabellen verteilt sind.

Wichtige JOIN-Arten

JOIN-Art	Bedeutung
INNER JOIN	nur passende Datensätze aus beiden Tabellen
LEFT JOIN	alle Datensätze der linken Tabelle
RIGHT JOIN	alle Datensätze der rechten Tabelle
FULL OUTER JOIN	alle Datensätze aus beiden Tabellen

INNER JOIN

Gibt nur Datensätze aus, bei denen die Join-Bedingung in beiden Tabellen erfüllt ist.

```
SELECT kunde.name, bestellung.bestellnr
FROM kunde
INNER JOIN bestellung
ON kunde.kundenr = bestellung.kundenr;
```

LEFT JOIN

Gibt alle Datensätze der linken Tabelle aus.

Fehlende passende Datensätze der rechten Tabelle werden mit **NULL** dargestellt.

```
SELECT kunde.name, bestellung.bestellnr
FROM kunde
LEFT JOIN bestellung
ON kunde.kundenr = bestellung.kundenr;
```

Grundregel

JOINS werden meistens über **Primärschlüssel** und **Fremdschlüssel** verbunden.

Merksatz

INNER JOIN = nur passende Datensätze

LEFT JOIN = alle Datensätze links + passende rechts

Prüfungshinweis

In IHK-Aufgaben ist meist entscheidend, ob auch Datensätze ohne passende Beziehung angezeigt werden sollen.

Typische Fehler:

- INNER JOIN und LEFT JOIN verwechseln.
 - JOIN-Bedingung vergessen.
 - falsche Schlüssel verbinden.
 - linke und rechte Tabelle vertauschen.
 - NULL-Werte bei LEFT JOIN nicht beachten.
-

Quellen

- IT-Handbuch für Fachinformatiker

8.4 Normalformen-Übersicht

Definition

Normalformen sind Regeln zur Strukturierung von Tabellen in relationalen Datenbanken.

Ziel ist es, Redundanzen, Inkonsistenzen und Anomalien zu vermeiden.

Übersicht

Normalform	Kernaussage
1. Normalform	alle Werte sind atomar
2. Normalform	jedes Nicht-Schlüsselattribut hängt vom gesamten Primärschlüssel ab
3. Normalform	keine Nicht-Schlüsselattribute hängen voneinander ab

1. Normalform

Eine Tabelle ist in der **1. Normalform**, wenn jedes Feld nur einen einzelnen Wert enthält.

Nicht erlaubt:

- mehrere Telefonnummern in einem Feld
- Listen in einer Zelle
- zusammengesetzte Werte in einem Feld

2. Normalform

Eine Tabelle ist in der **2. Normalform**, wenn sie in der 1. Normalform ist und jedes Nicht-Schlüsselattribut vom gesamten Primärschlüssel abhängig ist.

Wichtig besonders bei:

- zusammengesetzten Primärschlüsseln
- Zwischentabellen
- N:M-Beziehungen

3. Normalform

Eine Tabelle ist in der **3. Normalform**, wenn sie in der 2. Normalform ist und keine Nicht-Schlüsselattribute voneinander abhängig sind.

Merksatz

1NF = atomare Werte

2NF = vom ganzen Schlüssel abhängig

3NF = nur vom Schlüssel abhängig

Prüfungshinweis

In IHK-Aufgaben werden meistens die **1. bis 3. Normalform** abgefragt.

Typische Fehler:

- mehrere Werte in einem Feld zulassen
 - zusammengesetzten Primärschlüssel nicht beachten
 - abhängige Attribute in derselben Tabelle lassen
 - Normalisierung nur als Speicherplatzersparnis verstehen
-

Quellen

- IT-Handbuch für Fachinformatiker

8.5 Häufige Verwechslungen

Definition

Häufige Verwechslungen sind Begriffe, die in Datenbankaufgaben ähnlich wirken, aber fachlich unterschiedliche Bedeutungen haben.

Grundbegriffe

Begriff 1	Begriff 2	Unterschied
Datenbank	DBMS	Datenbank enthält Daten, DBMS verwaltet Daten
Tabelle	Datensatz	Tabelle enthält mehrere Datensätze
Datensatz	Feld	Datensatz = Zeile, Feld = einzelner Wert
Attribut	Datensatz	Attribut = Spalte, Datensatz = Zeile

Schlüssel

Begriff 1	Begriff 2	Unterschied
Primärschlüssel	Fremdschlüssel	Primärschlüssel identifiziert, Fremdschlüssel verweist
Primärschlüssel	Attribut	Primärschlüssel ist ein besonderes Attribut
Fremdschlüssel	Beziehung	Fremdschlüssel setzt eine Beziehung technisch um

Datenmodellierung

Begriff 1	Begriff 2	Unterschied
Entität	Attribut	Entität = Objekt, Attribut = Eigenschaft
Entität	Entitätstyp	Entität = einzelnes Objekt, Entitätstyp = Gruppe gleichartiger Objekte
Beziehung	Kardinalität	Beziehung verbindet Entitäten, Kardinalität beschreibt die Anzahl
N:M-Beziehung	Zwischentabelle	N:M wird durch eine Zwischentabelle aufgelöst

SQL

Begriff 1	Begriff 2	Unterschied
WHERE	HAVING	WHERE filtert Datensätze, HAVING filtert Gruppen
GROUP BY	ORDER BY	GROUP BY gruppiert, ORDER BY sortiert
COUNT	SUM	COUNT zählt Datensätze, SUM addiert Werte
INNER JOIN	LEFT JOIN	INNER JOIN nur passende, LEFT JOIN alle linken Datensätze
DELETE	DROP	DELETE löscht Datensätze, DROP löscht Tabellen oder Datenbankobjekte

Normalisierung

Begriff 1	Begriff 2	Unterschied
Redundanz	Inkonsistenz	Redundanz = doppelte Daten, Inkonsistenz = widersprüchliche Daten
1. Normalform	2. Normalform	1NF = atomar, 2NF = abhängig vom ganzen Schlüssel
2. Normalform	3. Normalform	2NF betrifft Schlüsselabhängigkeit, 3NF transitive Abhängigkeiten
Normalisierung	Denormalisierung	Normalisierung reduziert Redundanz, Denormalisierung nimmt Redundanz bewusst in Kauf

Merksatz

Viele Datenbankfehler entstehen durch verwechselte Begriffe.

Prüfungshinweis

In IHK-Aufgaben immer genau lesen, ob nach **Modellierung**, **Schlüsseln**, **SQL-Abfrage** oder **Normalisierung** gefragt wird.

Typische Fehler:

- Tabelle und Datensatz verwechseln.
- Primärschlüssel und Fremdschlüssel verwechseln.

- WHERE und HAVING verwechseln.
 - GROUP BY und ORDER BY verwechseln.
 - N:M-Beziehung ohne Zwischentabelle umsetzen.
-

Quellen

- IT-Handbuch für Fachinformatiker