**Business-Case 1: Kinoprogramm verwalten & anzeigen**

Ziel: Das Kino soll neue Filme und Vorstellungen einpflegen können und jederzeit das aktuelle Programm abrufen.

Datenerfassung:

- Einfügen eines neuen Films in die Film-Tabelle.

- Erstellen einer passenden Vorstellung mit Saal und Uhrzeit.

Informationsgewinnung:

- Abfrage: Welche Filme laufen heute mit Uhrzeit und Saal?

sql

SELECT f.Titel, v.Startzeit, v.Endzeit, s.Name AS Saal

FROM Vorstellung v

JOIN Film f ON f.FilmID = v.FilmID

JOIN Saal s ON s.SaalID = v.SaalID

WHERE TRUNC(v.Startzeit) = TRUNC(SYSDATE)

ORDER BY v.Startzeit;

**Business-Case 2: Ticketverkauf inkl. Snacks und Zahlungen**

Ziel: Kunden sollen Tickets buchen, Snacks kaufen und bezahlen können. Das Kino soll den Snackverkauf nachvollziehen können.

Datenerfassung:

- Reservierung/Ticket für einen Kunden anlegen

- Zahlung erfassen

- Snacks mit Menge dem Ticket zuordnen

Informationsgewinnung:

- Abfrage: Welche Snacks wurden zu welchen Vorstellungen verkauft (inkl. Menge)?

sql

SELECT f.Titel, v.Startzeit, sn.Name AS Snack, SUM(ts.Menge) AS GesamtMenge

FROM Ticket t

JOIN Vorstellung v ON t.VorstellungID = v.VorstellungID

JOIN Film f ON f.FilmID = v.FilmID

JOIN Ticket\_Snack ts ON ts.TicketID = t.TicketID

JOIN Snack sn ON sn.SnackID = ts.SnackID

GROUP BY f.Titel, v.Startzeit, sn.Name

ORDER BY f.Titel, v.Startzeit;

**Business-Case 3: Rabattaktionen verwalten & auswerten**

Ziel: Das Kino will bestimmte Filme für begrenzte Zeit mit Rabatten versehen – und sehen, welche aktuell laufen.

Datenerfassung:

- Rabattaktion erstellen mit Zeitspanne

- Rabattaktion einem Film zuweisen

Informationsgewinnung:

- Abfrage: Welche Filme haben derzeit gültige Rabattaktionen?

sql

SELECT f.Titel, r.Name AS Rabattaktion, r.Prozentsatz

FROM Rabattaktion r

JOIN Film f ON r.FilmID = f.FilmID

WHERE TRUNC(SYSDATE) BETWEEN r.Startdatum AND r.Enddatum;