

01 | Pizza-Delivery Order Management

- a) Ermitteln Sie alle Bestellungen eines bestimmten Tages, die einen Gesamtpreis von über 30 Euro haben und sich im Status OutForDelivery befinden. Verwenden Sie Where-Klauseln zur Filterung nach Datum, Preis und Status, und projizieren Sie die Ergebnisse auf ein anonymes Objekt, das Bestellnummer, Kundenname und Lieferadresse enthält. Sortieren Sie die Ergebnisse absteigend nach Bestellzeit.
- b) Gruppieren Sie alle Bestellungen nach Postleitzahl und berechnen Sie für jede Postleitzahl die Anzahl der Bestellungen, den durchschnittlichen Bestellwert und die Gesamtsumme aller Bestellungen. Verwenden Sie GroupBy in Kombination mit Aggregatfunktionen wie Count, Average und Sum. Filtern Sie anschließend nur die Postleitzahlen heraus, die mehr als 5 Bestellungen haben.
- c) Finden Sie die fünf beliebtesten Toppings über alle Bestellungen hinweg. Verwenden Sie SelectMany, um alle Toppings aus allen Bestellungen zu extrahieren, gruppieren Sie nach Topping-Namen, zählen Sie die Häufigkeit mit Count und sortieren Sie absteigend. Nehmen Sie dann die Top 5 mit Take und geben Sie sowohl den Topping-Namen als auch die Anzahl der Bestellungen aus.
- d) Erstellen Sie eine Liste aller Kunden, die in den letzten 30 Tagen mindestens drei Bestellungen aufgegeben haben und deren durchschnittlicher Bestellwert über 25 Euro liegt. Verwenden Sie Where zur zeitlichen Filterung, GroupBy zur Gruppierung nach Kundenname, Having-ähnliche Bedingungen mit Where nach dem GroupBy und Select zur Projektion der Kundendaten mit Bestellstatistiken.
- e) Kombinieren Sie die Bestelldaten mit einer Fahrerliste und ermitteln Sie für jeden Fahrer die Anzahl der ausgelieferten Bestellungen, die durchschnittliche Lieferzeit und die längste Lieferzeit des Tages. Verwenden Sie Join oder GroupJoin, um Bestellungen mit Fahrern zu verknüpfen, und nutzen Sie Aggregatfunktionen sowie OrderBy zur Sortierung nach Anzahl der Lieferungen.

02 | Hotel Reservation System

- a) Suchen Sie alle verfügbaren Zimmer für einen bestimmten Zeitraum, die eine bestimmte Zimmerkategorie haben und deren Preis innerhalb eines definierten Budgets liegt. Verwenden Sie Where-Klauseln mit Datumsvergleichen, Enum-Vergleichen und Preisfiltern. Projizieren Sie die Ergebnisse auf ein anonymes Objekt mit Zimmernummer, Typ, Preis und Etage, sortiert nach Preis aufsteigend.
- b) Berechnen Sie die Auslastungsstatistik pro Zimmerkategorie für den aktuellen Monat. Gruppieren Sie alle Buchungen nach RoomType, zählen Sie die Anzahl der Buchungstage, berechnen Sie die durchschnittliche Aufenthaltsdauer und ermitteln Sie den Gesamtumsatz pro Kategorie. Verwenden Sie GroupBy mit mehreren Aggregatfunktionen und Select für die Ergebnisprojektion.

- c) Finden Sie alle Gäste, die mehr als zweimal im Hotel übernachtet haben, und listen Sie deren Gesamtanzahl an Übernachtungen, den durchschnittlichen Ausgabenbetrag und das Datum der letzten Buchung auf. Verwenden Sie GroupBy zur Gruppierung nach Gast-ID oder E-Mail, Where zur Filterung nach Buchungsanzahl, und Select mit Aggregatfunktionen wie Count, Average und Max.
- d) Erstellen Sie einen Bericht über alle Buchungen mit Check-in am heutigen Tag, sortiert nach Check-in-Zeit. Verknüpfen Sie die Buchungsdaten mit den Zimmerdaten und Gästeden mittels Join-Operationen. Projizieren Sie auf ein umfassendes Objekt mit Gastname, Zimmernummer, Zimmertyp, Check-in-Zeit und speziellen Anforderungen, und verwenden Sie OrderBy für die zeitliche Sortierung.
- e) Identifizieren Sie Zimmer, die in den nächsten 7 Tagen niemals gebucht sind, um diese für Wartungsarbeiten einplanen zu können. Verwenden Sie einen Left Join oder Except-Operator, um alle Zimmer zu finden, die nicht in der gefilterten Buchungsliste für den Zeitraum erscheinen. Kombinieren Sie dies mit Where-Klauseln für den Datumszeitraum und Select für die Ausgabe relevanter Zimmerinformationen.

03 | Library Management System

- a) Finden Sie alle Bücher einer bestimmten Kategorie, die aktuell verfügbar sind und nach dem Jahr 2020 veröffentlicht wurden. Verwenden Sie Where-Klauseln zur Filterung nach BookCategory-Enum, LoanStatus und Veröffentlichungsdatum. Sortieren Sie die Ergebnisse alphabetisch nach Titel und projizieren Sie auf ein anonymes Objekt mit Titel, Autor, ISBN und Standort.
- b) Erstellen Sie eine Übersicht aller überfälligen Ausleihen, gruppiert nach Mitglied. Berechnen Sie für jedes Mitglied die Anzahl überfälliger Bücher, die Gesamtsumme der Mahngebühren und die durchschnittliche Anzahl überfälliger Tage. Verwenden Sie Where zur Identifikation überfälliger Ausleihen durch Datumsvergleich, GroupBy zur Gruppierung nach MemberId und Aggregatfunktionen für die Berechnungen.
- c) Ermitteln Sie die Top 10 der am häufigsten ausgeliehenen Bücher der letzten 6 Monate. Verwenden Sie Where zur zeitlichen Eingrenzung, GroupBy zur Gruppierung nach Book-Objekt oder ISBN, Count zur Zählung der Ausleihen, OrderByDescending zur Sortierung und Take zur Begrenzung auf 10 Ergebnisse. Projizieren Sie Titel, Autor, Kategorie und Ausleihanzahl.
- d) Finden Sie alle Mitglieder, die im letzten Jahr mehr als 15 Bücher ausgeliehen haben, aber aktuell keine aktive Ausleihe haben. Verwenden Sie zwei separate Abfragen oder eine komplexe Where-Klausel: eine zum Zählen der historischen Ausleihen mit GroupBy und Having-ähnlicher Filterung, und eine zum Prüfen des aktuellen Ausleihstatus. Kombinieren Sie beide mit einem Join oder verschachtelten Where-Klauseln.
- e) Erstellen Sie eine Autorenliste mit der Anzahl ihrer Bücher in der Bibliothek, der durchschnittlichen Bewertung (falls vorhanden) und der Gesamtanzahl der Ausleihen aller ihrer Bücher. Verwenden Sie SelectMany oder String-Operationen zur Extraktion von Autorennamen, GroupBy zur Gruppierung nach Autor, und Join-Operationen zur Verknüpfung mit Ausleihdaten. Sortieren Sie nach Gesamtausleihanzahl absteigend.

- a) Listen Sie alle Mitglieder auf, deren Mitgliedschaft in den nächsten 14 Tagen abläuft und die einen Annual- oder Quarterly-Plan haben. Verwenden Sie Where-Klauseln zur Filterung nach Ablaufdatum und MembershipPlan-Enum. Projizieren Sie auf ein Objekt mit Mitgliedsnummer, Name, Ablaufdatum und Kontaktinformationen, sortiert nach Ablaufdatum aufsteigend.
- b) Berechnen Sie die Auslastungsstatistik für jeden Kurstyp der letzten 30 Tage. Gruppieren Sie alle Kursbuchungen nach ClassType, zählen Sie die Anzahl der Teilnehmer, berechnen Sie die durchschnittliche Teilnehmerzahl pro Kurs und ermitteln Sie die Spitzenauslastung. Verwenden Sie GroupBy, Count, Average und Max in Kombination mit Where zur zeitlichen Eingrenzung.
- c) Identifizieren Sie die zehn aktivsten Mitglieder basierend auf der Anzahl besuchter Kurse im aktuellen Monat. Verwenden Sie Where zur Filterung nach Datum, GroupBy zur Gruppierung nach MemberId, Count zur Zählung der Kursbuchungen, OrderByDescending zur Sortierung und Take für die Top 10. Projizieren Sie Mitgliedername, Mitgliedschaftstyp und Anzahl besuchter Kurse.
- d) Finden Sie alle Kurse, die in der nächsten Woche stattfinden und noch freie Plätze haben. Verknüpfen Sie die Kursplanung mit den Buchungsdaten mittels GroupJoin oder Left Join, berechnen Sie die Differenz zwischen maximaler Kapazität und aktuellen Buchungen, und filtern Sie mit Where nach Verfügbarkeit und Zeitraum. Sortieren Sie nach Datum und Uhrzeit.
- e) Erstellen Sie einen Umsatzbericht, der die Einnahmen pro Mitgliedschaftsplan für das laufende Quartal zeigt. Gruppieren Sie alle Zahlungen nach MembershipPlan, summieren Sie die Beträge mit Sum, zählen Sie die Anzahl der Mitglieder pro Plan und berechnen Sie den durchschnittlichen Umsatz pro Mitglied. Verwenden Sie Where zur zeitlichen Eingrenzung und OrderByDescending zur Sortierung nach Gesamtumsatz.

05 | Restaurant Table Reservation System

- a) Finden Sie alle verfügbaren Tische für eine bestimmte Zeit und Personenanzahl während der Dinner-Periode. Verwenden Sie Where-Klauseln zur Filterung nach TableSize-Enum (passend zur Personenanzahl), MealPeriod-Enum und Verfügbarkeit durch Vergleich mit bestehenden Reservierungen. Verwenden Sie Except oder einen Left Join, um bereits reservierte Tische auszuschließen.
- b) Erstellen Sie eine Statistik über No-Shows gruppiert nach Wochentag und Tageszeit. Verwenden Sie Where zur Filterung nach ReservationStatus.NoShow, extrahieren Sie Wochentag und MealPeriod, gruppieren Sie mit GroupBy nach beiden Dimensionen und zählen Sie die Vorkommen mit Count. Projizieren Sie auf ein aussagekräftiges Ergebnisobjekt.
- c) Listen Sie alle Stammgäste auf, die in den letzten 3 Monaten mindestens 5 Reservierungen hatten, sortiert nach Anzahl der Besuche. Verwenden Sie Where zur zeitlichen Filterung, GroupBy zur Gruppierung nach Gastname oder Telefonnummer, Having-ähnliche Filterung mit Where nach dem GroupBy, Count zur Zählung und OrderByDescending zur Sortierung.

- d) Ermitteln Sie die durchschnittliche Auslastung pro Tischgröße für verschiedene Wochentage. Verknüpfen Sie Reservierungen mit Tischdaten mittels Join, gruppieren Sie nach TableSize und Wochentag, berechnen Sie die Anzahl der Reservierungen und teilen Sie durch die Anzahl verfügbarer Tische dieser Größe. Verwenden Sie komplexe GroupBy-Klauseln und Aggregatfunktionen.
- e) Finden Sie alle Reservierungen für den heutigen Abend, die spezielle Anforderungen haben (Allergieinformationen, Kinderstuhl, etc.). Verwenden Sie Where zur Filterung nach Datum, MealPeriod und nicht-leeren Notizen oder speziellen Flags. Verknüpfen Sie mit Gästedaten mittels Join und projizieren Sie auf ein vollständiges Reservierungsobjekt mit allen relevanten Details, sortiert nach Reservierungszeit.

06 | Car Rental Management System

- a) Suchen Sie alle verfügbaren Fahrzeuge einer bestimmten Kategorie innerhalb einer Preisspanne für einen gewünschten Mietzeitraum. Verwenden Sie Where-Klauseln zur Filterung nach VehicleCategory-Enum, Preisgrenzen und RentalStatus.Available. Prüfen Sie mit einer Unterabfrage oder Left Join, dass keine überlappenden Mietverträge existieren. Sortieren Sie nach Preis aufsteigend.
- b) Erstellen Sie einen Bericht über die Flottenauslastung pro Fahrzeugkategorie für das letzte Quartal. Gruppieren Sie alle Mietverträge nach VehicleCategory, berechnen Sie die Gesamtanzahl der Miettage mit Sum, die durchschnittliche Mietdauer mit Average und den Gesamtumsatz pro Kategorie. Verwenden Sie Where zur zeitlichen Eingrenzung und Aggregatfunktionen.
- c) Identifizieren Sie die Top 5 Fahrzeuge mit dem höchsten Umsatz in diesem Jahr. Verknüpfen Sie Fahrzeugdaten mit Mietvertragsdaten mittels Join, gruppieren Sie nach Vehicle-Objekt oder Kennzeichen, summieren Sie die Mieteinnahmen mit Sum, sortieren Sie mit OrderByDescending und nehmen Sie die Top 5 mit Take. Projizieren Sie Kennzeichen, Modell, Kategorie und Gesamtumsatz.
- d) Finden Sie alle Fahrzeuge, die mehr als 100.000 Kilometer gefahren sind oder in den nächsten 14 Tagen zur Wartung fällig sind. Verwenden Sie Where mit OR-Bedingungen zur Filterung nach Kilometerstand oder Wartungsdatum. Verknüpfen Sie mit der Wartungswarteschlange mittels Join oder Contains-Operationen und projizieren Sie auf ein Wartungsobjekt mit Priorität.
- e) Erstellen Sie eine Liste aller Kunden mit überfälligen Rückgaben. Verwenden Sie Where zur Identifikation von Mietverträgen, deren geplantes Rückgabedatum überschritten ist und der Status noch Rented ist. Verknüpfen Sie mit Kundendaten mittels Join, berechnen Sie die Anzahl überfälliger Tage und die anfallenden Zusatzgebühren. Sortieren Sie nach Anzahl der überfälligen Tage absteigend.

- a) Listen Sie alle Termine für einen bestimmten Arzt an einem bestimmten Tag auf, sortiert nach Uhrzeit. Filtern Sie mit Where nach Doctor-ID und Datum, sortieren Sie mit OrderBy nach Terminzeit und projizieren Sie auf ein Objekt mit Patientennamen, Terminzeit, AppointmentType und geschätzter Dauer. Schließen Sie stornierte Termine aus.
- b) Erstellen Sie eine Statistik über die durchschnittliche Wartezeit pro Abteilung im letzten Monat. Gruppieren Sie alle abgeschlossenen Termine nach Department-Enum, berechnen Sie die durchschnittliche Differenz zwischen geplanter und tatsächlicher Startzeit mit Average, und zählen Sie die Anzahl der Termine mit Count. Verwenden Sie Where zur zeitlichen und Status-Filterung.
- c) Identifizieren Sie alle Patienten mit mehr als drei Terminen im letzten Jahr, gruppiert nach Abteilung. Verwenden Sie Where zur zeitlichen Filterung, GroupBy zur Gruppierung nach PatientID und Department, Having-ähnliche Filterung nach dem GroupBy und Count zur Zählung. Verknüpfen Sie mit Patientendaten mittels Join für vollständige Informationen.
- d) Finden Sie alle verfügbaren Zeitslots für Emergency-Termine in den nächsten 24 Stunden über alle Abteilungen hinweg. Generieren Sie eine Liste möglicher Zeitslots, verwenden Sie Except oder einen komplexen Where mit Unterabfrage, um bereits belegte Slots auszuschließen. Gruppieren Sie die Ergebnisse nach Department und sortieren Sie nach Zeit aufsteigend.
- e) Erstellen Sie eine No-Show-Analyse, die zeigt, welche Patienten in den letzten 6 Monaten mehr als zweimal nicht zu Terminen erschienen sind. Verwenden Sie Where zur Filterung nach AppointmentStatus.NoShow und Zeitraum, GroupBy zur Gruppierung nach PatientID, Having-ähnliche Filterung nach Anzahl, und Join zur Verknüpfung mit Patientendetails. Berechnen Sie zusätzlich die geschätzten Verlustkosten pro Patient.

- a) Finden Sie alle verfügbaren Tickets für ein bestimmtes Event in einer bestimmten Preiskategorie. Verwenden Sie Where-Klauseln zur Filterung nach Event-ID, TicketStatus.Available, TicketType-Enum und Preisspanne. Projizieren Sie auf ein Objekt mit Sitznummer, Sektion, Preis und Ticket-ID, sortiert nach Sitznummer aufsteigend.
- b) Erstellen Sie eine Verkaufsstatistik pro EventCategory für das laufende Jahr. Gruppieren Sie alle verkauften Tickets nach EventCategory-Enum, zählen Sie die Anzahl mit Count, summieren Sie den Umsatz mit Sum, berechnen Sie den durchschnittlichen Ticketpreis mit Average und ermitteln Sie das teuerste verkaufte Ticket mit Max. Verwenden Sie Where zur zeitlichen Eingrenzung.
- c) Identifizieren Sie die Top 10 Events mit den höchsten Verkaufszahlen in den letzten 3 Monaten. Verwenden Sie Where zur zeitlichen Filterung, GroupBy zur Gruppierung nach Event-ID, Count zur Zählung der verkauften Tickets, OrderByDescending zur Sortierung und Take für die Top 10. Verknüpfen Sie mit Event-Details mittels Join für vollständige Informationen wie Name, Datum und Kategorie.
- d) Listen Sie alle Kunden auf, die in diesem Monat Tickets für mehr als drei verschiedene Events gekauft haben. Verwenden Sie Where zur zeitlichen Filterung, GroupBy zur Gruppierung nach Customer-ID, verwenden Sie Distinct oder GroupBy mit Count auf Event-IDs innerhalb jeder Kundengruppe, und filtern Sie mit Where nach der Anzahl unterschiedlicher Events. Verknüpfen Sie mit Kundendaten für vollständige Profile.
- e) Erstellen Sie einen Bericht über die Nutzung von Gutscheincodes, gruppiert nach Code und EventCategory. Filtern Sie mit Where nach Tickets, bei denen ein Gutscheincode verwendet wurde, gruppieren Sie mit GroupBy nach Code und EventCategory, zählen Sie die Nutzungen mit Count, summieren Sie die gewährten Rabatte mit Sum und berechnen Sie den durchschnittlichen Rabattbetrag mit Average. Sortieren Sie nach Gesamtrabatt absteigend, um die wirkungsvollsten Gutscheine zu identifizieren.

- a) Ermitteln Sie alle Kurse einer bestimmten Schwierigkeitsstufe, die in den letzten 90 Tagen veröffentlicht wurden und eine durchschnittliche Bewertung von mindestens 4.0 Sternen haben. Verwenden Sie Where-Klauseln zur Filterung nach DifficultyLevel-Enum, Veröffentlichungsdatum und Rating. Projizieren Sie auf ein anonymes Objekt mit Kurstitel, Instructor-Name, Preis, durchschnittlicher Bewertung und Anzahl der Einschreibungen, sortiert nach Bewertung absteigend.
- b) Gruppieren Sie alle Kurseinschreibungen nach CourseCategory und berechnen Sie für jede Kategorie die Gesamtanzahl der Einschreibungen, die durchschnittliche Kursabschlussrate (Percentage Completed), den Gesamtumsatz und die durchschnittliche Kursdauer. Verwenden Sie GroupBy in Kombination mit Count, Average und Sum. Filtern Sie anschließend nur die Kategorien heraus, die mehr als 100 aktive Einschreibungen haben, und sortieren Sie nach Umsatz absteigend.
- c) Finden Sie die zehn beliebtesten Kurse basierend auf der Anzahl abgeschlossener Einschreibungen in den letzten 6 Monaten. Verwenden Sie Where zur zeitlichen Filterung und Status-Filterung (CompletionStatus.Completed), GroupBy zur Gruppierung nach Course-ID, Count zur Zählung, OrderByDescending zur Sortierung und Take für die Top 10. Verknüpfen Sie mit Kursdaten mittels Join und projizieren Sie Kurstitel, Instructor, Kategorie, Anzahl der Abschlüsse und durchschnittliche Bewertung.
- d) Erstellen Sie eine Liste aller Studenten, die im aktuellen Quartal mindestens fünf Kurse abgeschlossen haben und deren durchschnittliche Abschlusszeit unter 80% der empfohlenen Kursdauer lag. Verwenden Sie Where zur zeitlichen Filterung und Status-Prüfung, GroupBy zur Gruppierung nach Student-ID, Having-ähnliche Bedingungen mit Where nach dem GroupBy zur Filterung nach Anzahl und Geschwindigkeit. Berechnen Sie zusätzlich die durchschnittliche Lerngeschwindigkeit und Gesamtlernzeit mit Select und Aggregatfunktionen.
- e) Kombinieren Sie Kursdaten mit Instructor-Bewertungen und ermitteln Sie für jeden Instructor die Anzahl veröffentlichter Kurse, die durchschnittliche Kursbewertung über alle Kurse, die Gesamtanzahl der Studenten, den Gesamtumsatz und die höchste Einzelbewertung. Verwenden Sie Join zur Verknüpfung von Kursen mit Instructors, GroupBy zur Gruppierung nach Instructor-ID, und nutzen Sie Count, Average, Sum und Max. Filtern Sie Instructors mit mindestens drei aktiven Kursen und sortieren Sie nach Gesamtumsatz absteigend.

- a) Suchen Sie alle Produkte einer bestimmten Kategorie, deren Lagerbestand unter dem Mindestbestand liegt und die in den letzten 30 Tagen mindestens einmal verkauft wurden. Verwenden Sie Where-Klauseln zur Filterung nach ProductCategory-Enum, Bestandsvergleich ($\text{CurrentStock} < \text{MinimumStock}$) und Verkaufsdatum. Projizieren Sie auf ein Objekt mit Produktname, SKU, aktueller Bestand, Mindestbestand, Differenz und letztes Verkaufsdatum, sortiert nach Bestandsdifferenz aufsteigend (kritischste zuerst).
- b) Berechnen Sie die Lagerumschlagsrate pro Produktkategorie für das letzte Halbjahr. Gruppieren Sie alle Verkaufstransaktionen nach ProductCategory, summieren Sie die verkauften Mengen mit Sum, berechnen Sie den durchschnittlichen Lagerbestand über den Zeitraum mit Average und ermitteln Sie die Umschlagsrate (verkaufte Menge / durchschnittlicher Bestand). Verwenden Sie GroupBy mit komplexen Aggregatfunktionen, Where zur zeitlichen Eingrenzung und OrderByDescending zur Sortierung nach Umschlagsrate. Projizieren Sie Kategorie, Gesamtverkäufe, durchschnittlicher Bestand und Umschlagsrate.
- c) Identifizieren Sie die 15 profitabelsten Produkte basierend auf der Gesamtmarge (Verkaufspreis minus Einkaufspreis multipliziert mit verkaufter Menge) im aktuellen Quartal. Verwenden Sie Where zur zeitlichen Filterung, GroupBy zur Gruppierung nach Product-ID oder SKU, Select zur Berechnung der Marge pro Verkauf, Sum zur Gesamtmarge, OrderByDescending zur Sortierung und Take für die Top 15. Verknüpfen Sie mit Produktdaten mittels Join für vollständige Informationen wie Name, Kategorie und aktuellen Bestand.
- d) Finden Sie alle Produkte mit Ablaufdatum in den nächsten 30 Tagen, gruppiert nach WarehouseLocation. Verwenden Sie Where zur Filterung nach Ablaufdatum und StockStatus (nicht Discontinued), GroupBy zur Gruppierung nach Lagerstandort, Count zur Zählung pro Standort und Sum zur Berechnung des potenziellen Verlustwerts. Projizieren Sie auf ein Objekt mit Standort, Anzahl ablaufender Produkte, Gesamtwert und Liste der kritischsten Artikel, sortiert nach Verlustwert absteigend.
- e) Erstellen Sie einen Nachbestellbericht, der für jede Produktkategorie die Anzahl der Artikel unter Mindestbestand, den durchschnittlichen Fehlbestand (Mindestbestand minus aktueller Bestand), die geschätzten Nachbestellkosten und die Dringlichkeit basierend auf Verkaufsgeschwindigkeit zeigt. Verwenden Sie Where zur Filterung nach LowStock oder Out OfStock Status, GroupBy zur Gruppierung nach ProductCategory, komplexe Berechnungen mit Select für Fehlbestand und Kosten, und OrderBy zur Sortierung nach Dringlichkeit. Verknüpfen Sie mit Verkaufsdaten mittels Join zur Ermittlung der täglichen Verkaufsrate.

- a) Listen Sie alle Sendungen auf, die heute zur Auslieferung fällig sind, aber noch im Status InTransit sind und Express- oder SameDay-Priorität haben. Verwenden Sie Where-Klauseln zur Filterung nach geplanter Lieferzeit (heute), ShipmentStatus und ShipmentPriority-Enum. Projizieren Sie auf ein Objekt mit Tracking-Code, Empfängername, Lieferadresse, Priorität und Verspätung in Stunden, sortiert nach Verspätung absteigend.
- b) Gruppieren Sie alle Sendungen der letzten 30 Tage nach DeliveryMethod und berechnen Sie für jede Versandart die Gesamtanzahl der Sendungen, die durchschnittliche Lieferzeit, die Erfolgsquote (Delivered / Gesamt) und die durchschnittlichen Versandkosten. Verwenden Sie GroupBy in Kombination mit Count, Average und Select für komplexe Berechnungen. Filtern Sie anschließend nur Versandarten mit mehr als 50 Sendungen und sortieren Sie nach Erfolgsquote absteigend.
- c) Finden Sie die zehn häufigsten Lieferziele (Städte) in diesem Monat. Verwenden Sie Where zur zeitlichen Filterung, SelectMany oder String-Operationen zur Extraktion der Stadt aus Lieferadressen, GroupBy zur Gruppierung nach Stadt, Count zur Zählung, OrderByDescending zur Sortierung und Take für die Top 10. Projizieren Sie Stadtname, Anzahl der Lieferungen, durchschnittliche Lieferzeit und Gesamtversandvolumen.
- d) Erstellen Sie eine Liste aller Sendungen, die in den letzten 14 Tagen fehlgeschlagen oder zurückgesendet wurden, gruppiert nach Fehlergrund. Verwenden Sie Where zur zeitlichen Filterung und Status-Filterung (Failed oder Returned), GroupBy zur Gruppierung nach Fehlercode oder Rücksendungsgrund, Count zur Zählung pro Grund und Sum zur Berechnung der entstandenen Kosten. Verknüpfen Sie mit Sendungsdetails mittels Join für vollständige Informationen, sortiert nach Häufigkeit absteigend.
- e) Kombinieren Sie Sendungsdaten mit Fahrerdaten und ermitteln Sie für jeden Fahrer die Anzahl der heute ausgelieferten Sendungen, die durchschnittliche Anzahl der Stopps pro Tour, die Gesamtstrecke in Kilometern und die Pünktlichkeitsquote (rechtzeitig zugestellt / gesamt). Verwenden Sie Join oder GroupJoin zur Verknüpfung von Sendungen mit Fahrern, GroupBy zur Gruppierung nach Fahrer-ID, komplexe Aggregatfunktionen zur Berechnung der Metriken und OrderByDescending zur Sortierung nach Anzahl der Lieferungen. Filtern Sie nur aktive Fahrer mit mindestens einer Lieferung heute.

- a) Suchen Sie alle verfügbaren Immobilien eines bestimmten Typs in einer definierten Preisspanne, die mindestens eine bestimmte Anzahl von Zimmern haben und vollständig oder teilweise möbliert sind. Verwenden Sie Where-Klauseln zur Filterung nach PropertyTypeEnum, PropertyStatus.Available, Mietpreis, Zimmerzahl und FurnishingTypeEnum. Projizieren Sie auf ein Objekt mit Adresse, Quadratmeter, Zimmeranzahl, Preis und Ausstattung, sortiert nach Preis pro Quadratmeter aufsteigend.
- b) Berechnen Sie die Vermietungsstatistik pro Stadtteil für das laufende Jahr. Gruppieren Sie alle Mietverträge nach Stadtteil (aus Adresse extrahieren), zählen Sie die Anzahl vermieteter Immobilien mit Count, berechnen Sie den durchschnittlichen Mietpreis mit Average, die durchschnittliche Vermietungsdauer mit Average und den Gesamtumsatz mit Sum. Verwenden Sie GroupBy mit mehreren Aggregatfunktionen, Where zur zeitlichen Eingrenzung und OrderByDescending zur Sortierung nach Gesamtumsatz.
- c) Identifizieren Sie die zehn profitabelsten Immobilien basierend auf dem Verhältnis von Mieteinnahmen zu Wartungskosten im letzten Jahr. Verwenden Sie Where zur zeitlichen Filterung, Join zur Verknüpfung von Immobilien mit Mietverträgen und Wartungsdaten, GroupBy zur Gruppierung nach Property-ID, Sum zur Berechnung von Gesamteinnahmen und -kosten, Select zur Berechnung der Profitabilität, OrderByDescending zur Sortierung und Take für die Top 10. Projizieren Sie Adresse, Immobilientyp, Gesamteinnahmen, Gesamtkosten und Profitrate.
- d) Finden Sie alle Mietverträge, die in den nächsten 60 Tagen auslaufen und noch keine Verlängerung oder Kündigung registriert haben. Verwenden Sie Where zur Filterung nach Vertragsende-Datum, LeaseType und Verlängerungs-/Kündigungsstatus (null oder leer). Verknüpfen Sie mit Immobilien- und Mieterdaten mittels Join, projizieren Sie auf ein vollständiges Objekt mit Mieter-Kontaktdaten, Immobilienadresse, Auslaufdatum und Vertragsbedingungen, sortiert nach Auslaufdatum aufsteigend.
- e) Erstellen Sie einen Wartungsbericht, der für jede Immobilie die Anzahl der Wartungsanfragen im letzten Quartal, die durchschnittliche Bearbeitungszeit, die Gesamtwartungskosten und die häufigsten Wartungsgründe zeigt. Verwenden Sie Where zur zeitlichen Filterung, GroupBy zur Gruppierung nach Property-ID, Count für Anzahl der Anfragen, Average für Bearbeitungszeit, Sum für Kosten und eine verschachtelte GroupBy-Abfrage zur Ermittlung des häufigsten Wartungsgrunds. Verknüpfen Sie mit Immobiliendaten mittels Join, filtern Sie Immobilien mit mehr als drei Anfragen und sortieren Sie nach Wartungskosten absteigend.