

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ
Π.Μ.Σ. Ανάπτυξη Λογισμικού και Νέφος

Διαχείριση και Αναλυτική Δεδομένων
στο Υπολογιστικό Νέφος

MongoDB

ΕΥΣΤΑΘΙΟΣ ΙΩΣΗΦΙΔΗΣ

AM: mai25017

Ερώτημα Α

1. Εισαγωγή

Αυτή η αναφορά παρουσιάζει δύο εναλλακτικούς σχεδιασμούς βάσεων δεδομένων MongoDB για ένα σύστημα ταξιδιωτικού γραφείου, ο καθένας βελτιστοποιημένος για διαφορετικά σύνολα ερωτημάτων (query patterns). Ο πρώτος σχεδιασμός (Σχεδιασμός Α) είναι βελτιστοποιημένος για ερωτήματα που αφορούν συγκεντρωτικές πληροφορίες και στατιστικά στοιχεία (queries τύπου α), ενώ ο δεύτερος σχεδιασμός (Σχεδιασμός Β) είναι βελτιστοποιημένος για ερωτήματα που επικεντρώνονται στην ανάκτηση δεδομένων για συγκεκριμένες οντότητες, όπως οι κρατήσεις ενός πελάτη ή οι εκδρομές ενός ξεναγού (queries τύπου β). Στόχος είναι η ανάδειξη των σχεδιαστικών επιλογών και των συνεπαγόμενων πλεονεκτημάτων και μειονεκτημάτων κάθε προσέγγισης στο πλαίσιο των απαιτήσεων του ταξιδιωτικού γραφείου.

2. Μεθοδολογία

Στην παρούσα ανάλυση, ακολουθήθηκε μια μεθοδολογία σχεδιασμού βάσεων δεδομένων MongoDB με γνώμονα την απόδοση συγκεκριμένων τύπων ερωτημάτων. Η διαδικασία περιλάμβανε τα εξής βήματα:

1. **Ανάλυση Απαιτήσεων:** Μελετήθηκαν τα δύο διακριτά σύνολα ερωτημάτων (α και β) που τέθηκαν ως απαιτήσεις για το σύστημα του ταξιδιωτικού γραφείου. Αυτά τα ερωτήματα καθόρισαν τα κριτήρια βελτιστοποίησης για κάθε σχεδιασμό.
2. **Προσδιορισμός Οντοτήτων:** Εντοπίστηκαν οι κύριες οντότητες του συστήματος: Πελάτες (Customers), Πτήσεις (Flights), Ξενοδοχεία (Hotels), Ενοικιάσεις Αυτοκινήτων (Car Rentals), Κρατήσεις (Bookings), Εκδρομές (Tours) και Ξεναγοί (Tour Guides).
3. **Σχεδιαστικές Προσεγγίσεις MongoDB:** Για την υλοποίηση των δύο εναλλακτικών σχεδιασμών, αξιοποιήθηκαν οι θεμελιώδεις τεχνικές μοντελοποίησης της MongoDB:
 - ο **Σχεσιακή Προσέγγιση (Referencing):** Χρησιμοποιήθηκαν αναφορές (ObjectIDs) για τη σύνδεση εγγράφων μεταξύ διαφορετικών συλλογών. Αυτή η προσέγγιση αποτέλεσε τη βάση για τον Σχεδιασμό Α, ο οποίος ευνοεί την ακεραιότητα και τη μείωση της επανάληψης δεδομένων, αλλά μπορεί να απαιτεί πολλαπλές αναζητήσεις (joins μέσω \$lookup) για τη σύνθεση πλήρους πληροφορίας.
 - ο **Ενσωμάτωση (Embedding):** Ενσωματώθηκαν σχετικά δεδομένα απευθείας μέσα στα έγγραφα γονείς. Αυτή η προσέγγιση υιοθετήθηκε εκτενώς στον Σχεδιασμό Β, με στόχο την ταχύτερη ανάκτηση δεδομένων για συγκεκριμένες οντότητες μέσω λιγότερων ή μίας μόνο ανάγνωσης, με πιθανό κόστος την επανάληψη δεδομένων και την πολυπλοκότητα στις ενημερώσεις.
4. **Καθορισμός Δομής Συλλογών και Εγγράφων:** Για κάθε σχεδιασμό (Α και Β), ορίστηκαν οι απαραίτητες συλλογές (collections) και η δομή των εγγράφων (documents) εντός αυτών, συμπεριλαμβάνοντας τα πεδία που κρίθηκαν χρήσιμα για την κάλυψη των λειτουργικών αναγκών και των ερωτημάτων. Υποθέσαμε τη χρήση κατάλληλων τύπων δεδομένων (π.χ., ISODate για ημερομηνίες, ObjectId για αναφορές).

5. **Αξιολόγηση Βάσει Ερωτημάτων:** Κάθε σχεδιασμός αξιολογήθηκε θεωρητικά ως προς την ευκολία και την απόδοση με την οποία μπορεί να απαντήσει στο αντίστοιχο σύνολο ερωτημάτων (α ή β) για το οποίο βελτιστοποιήθηκε, καθώς και στο εναλλακτικό σύνολο.

3. Αποτελέσματα: Ανάλυση Σχεδιασμών

Σε αυτή την ενότητα παρουσιάζονται αναλυτικά οι δύο προτεινόμενοι σχεδιασμοί. Για κάθε σχεδιασμό, περιγράφεται η δομή των βασικών συλλογών μέσω δειγμάτων εγγράφων και αναλύεται η απόδοσή του ως προς τα καθορισμένα σύνολα ερωτημάτων (α) και (β).

- **3.1 Σχεδιασμός A: Βελτιστοποίηση για Ερωτήματα (α)**

- *Φιλοσοφία:* Ο Σχεδιασμός A υιοθετεί μια προσέγγιση που μοιάζει περισσότερο με τις σχεσιακές βάσεις δεδομένων, χρησιμοποιώντας αναφορές (references) με τη μορφή ObjectIds για τη σύνδεση εγγράφων μεταξύ διαφορετικών, εξειδικευμένων συλλογών. Στόχος είναι η ελαχιστοποίηση της επανάληψης δεδομένων (normalization) και η διατήρηση της συνοχής. Αυτή η δομή είναι εγγενώς κατάλληλη για ερωτήματα που απαιτούν συγκεντρωτικές λειτουργίες (aggregation) ή φιλτράρισμα πάνω σε μεγάλους όγκους ομοειδών οντοτήτων (π.χ., όλες οι πτήσεις προς έναν προορισμό, όλα τα ξενοδοχεία σε ένα εύρος τιμών).
- *Δομή Συλλογών (customers, flights, hotels, car_rentals, tours, tour_guides, bookings):*

1. customers: Πληροφορίες πελατών.

```
{
  "_id": ObjectId("60f5a9f7e8b4a0b3e8f1b1b1"),
  "first_name": "Γιώργος",
  "last_name": "Παπαδόπουλος",
  "email": "mai24000@uom.edu.gr",
  "phone": "2311234567",
  "address": {
    "street": "Λεωφ. Αθηνών 10",
    "city": "Θεσσαλονίκη",
    "postal_code": "54242",
    "country": "Ελλάδα"
  },
  "registration_date": ISODate("2025-01-15T10:00:00Z")
  // Δεν ενσωματώνουμε κρατήσεις εδώ για να απαντήσουμε εύκολα στο α.1
  μέσω της συλλογής bookings
}
```

2. flights: Πληροφορίες πτήσεων.

```
{
  "_id": ObjectId("60f5aa3be8b4a0b3e8f1b1b2"),
  "airline": "Aegean Airlines",
```

```

"flight_number": "A3 501",
"departure": {
  "airport_code": "SKG",
  "city": "Θεσσαλονίκη",
  "country": "Ελλάδα",
  "datetime": ISODate("2025-01-20T14:30:00Z")
},
"arrival": {
  "airport_code": "ATH",
  "city": "Αθήνα",
  "country": "Ελλάδα",
  "datetime": ISODate("2025-01-20T15:15:00Z")
},
"price_eur": 80.50,
"available_seats": 120
}

```

3. hotels: Πληροφορίες ξενοδοχείων.

```

{
  "_id": ObjectId("60f5aa6ce8b4a0b3e8f1b1b3"),
  "name": "Hotel President",
  "city": "Αθήνα",
  "country": "Ελλάδα",
  "stars": 5,
  "address": "Λεωφ. Κηφισίας 43",
  "amenities": ["Pool", "WiFi", "Restaurant", "Spa"],
  "price_per_night_eur": 180.00,
  "contact_email": "president@president.gr"
}

```

4. car_rentals: Πληροφορίες εταιριών και διαθέσιμων αυτοκινήτων (απλοποιημένο).

```

{
  "_id": ObjectId("60f5aa9be8b4a0b3e8f1b1b4"),
  "company_name": "Hertz",
  "pickup_location": {
    "city": "Αθήνα",
    "point": "Ελ. Βενιζέλος (ATH)"
  },
  "car_model": "BYD",
  "category": "EV",
  "price_per_day_eur": 40.00,
  "available_from": ISODate("2025-01-20T00:00:00Z"),
  "available_until": ISODate("2025-01-23T23:59:59Z")
}

```

5. tours: Πληροφορίες εκδρομών.

```
{
  "_id": ObjectId("60f5aad5e8b4a0b3e8f1b1b5"),
  "name": "Ημερήσια Εκδρομή Ακρόπολη, Πλάκα, Μοναστηράκι",
  "destination_city": "Αθήνα",
  "duration_hours": 6,
  "description": "Εκδρομή στην Ακρόπολη, γεύμα στην Πλάκα, αγορές στο Μοναστηράκι",
  "price_eur": 35.00,
  "guide_id": ObjectId("60f5ab0fe8b4a0b3e8f1b1b6"), // Reference στον ξεναγό
  "scheduled_dates": [
    ISODate("2025-01-24T09:00:00Z"),
    ISODate("2025-01-25T09:00:00Z")
  ]
}
```

6. tour_guides: Πληροφορίες ξεναγών.

```
{
  "_id": ObjectId("60f5ab0fe8b4a0b3e8f1b1b6"),
  "first_name": "Γεωργία",
  "last_name": "Νικολάου",
  "languages": ["Ελληνικά", "Αγγλικά"],
  "specialization": ["Ιστορία", "Αρχαιολογία"],
  "contact_phone": "6999999999"
}
```

7. bookings: Κεντρική συλλογή κρατήσεων, συνδέει πελάτες με υπηρεσίες.

```
{
  "_id": ObjectId("60f5ab4ae8b4a0b3e8f1b1b7"),
  "customer_id": ObjectId("60f5a9f7e8b4a0b3e8f1b1b1"), // Reference στον πελάτη
  "booking_date": ISODate("2025-01-15T15:30:00Z"),
  "total_price_eur": 1150.50,
  "status": "Confirmed", // Π.χ., Pending, Confirmed, Cancelled
  "items": [ // Λίστα με τα στοιχεία της κράτησης
    {
      "type": "flight",
      "flight_id": ObjectId("60f5aa3be8b4a0b3e8f1b1b2"), // Reference στην πτήση
      "departure_date": ISODate("2025-01-20T14:30:00Z"),
      "passengers": 1,
      "price_paid_eur": 80.50
    },
    {

```

```

      "type": "hotel",
      "hotel_id": ObjectId("60f5aa6ce8b4a0b3e8f1b1b3"), // Reference στο
      ξενοδοχείο
      "check_in_date": ISODate("2025-01-20T15:00:00Z"),
      "check_out_date": ISODate("2025-01-25T11:00:00Z"),
      "rooms": 1,
      "guests": 1,
      "price_paid_eur": 900.00 // (5 νύχτες * 180)
    },
    {
      "type": "tour",
      "tour_id": ObjectId("60f5aad5e8b4a0b3e8f1b1b5"), // Reference στην
      εκδρομή
      "tour_date": ISODate("2025-01-24T09:00:00Z"),
      "participants": 1,
      "price_paid_eur": 35.00
    },
    {
      "type": "car",
      "rental_details_id": ObjectId("60f5aa9be8b4a0b3e8f1b1b4"), // Reference
      στην ενοικίαση
      "pickup_datetime": ISODate("2025-01-20T10:00:00Z"),
      "dropoff_datetime": ISODate("2025-01-23T10:00:00Z"),
      "days": 3,
      "price_paid_eur": 135.00 // (3 ημέρες * 40 + φόροι/ασφάλεια ίσως)
    }
  ]
}

```

Τεκμηρίωση Επιλογών (Σχεδιασμός Α):

Για Ερωτήματα (α):

- **α.1 (Πόσες κρατήσεις/πελάτη;):** Πολύ αποτελεσματικό μέσω aggregation στη συλλογή bookings με group by customer_id και count. `db.bookings.aggregate([{ $group: { _id: "$customer_id", count: { $sum: 1 } } }])`
- **α.2 (Κρατήσεις/χρονικό διάστημα;):** Αποτελεσματικό με απλό query (find) στη συλλογή bookings με φίλτρο στο πεδίο booking_date ή στα πεδία ημερομηνιών μέσα στο items array αν αφορά την περίοδο παροχής της υπηρεσίας. Π.χ., `db.bookings.find({ booking_date: { $gte: ISODate("..."), $lte: ISODate("...") } })`
- **α.3 (Πτήσεις/προορισμό/προέλευση/ημερομηνία;):** Πολύ αποτελεσματικό και γρήγορο με απλό query (find) στη συλλογή flights. Π.χ., `db.flights.find({ "arrival.city": "Σαντορίνη", "departure.datetime": { $gte: ISODate("..."), $lt: ISODate("...") } })`
- **α.4 (Ξενοδοχεία/εύρος τιμών;):** Πολύ αποτελεσματικό και γρήγορο με απλό query (find) στη συλλογή hotels. Π.χ., `db.hotels.find({ price_per_night_eur: { $gte: 100, $lte: 200 } })`
- **α.5 (Εκδρομές/πελάτη;):** Λίγο πιο σύνθετο αλλά εφικτό. Aggregation στη συλλογή bookings: `$match` για συγκεκριμένο customer_id, `$unwind` το items array, `$match` για

items.type: "tour", \$lookup στη συλλογή tours για να πάρουμε λεπτομέρειες.
db.bookings.aggregate([...])

Για Ερωτήματα (β):

- **β.1 (Κρατήσεις συγκεκριμένου πελάτη):** Απαιτεί τουλάχιστον δύο αναγνώσεις: μία για τα βασικά στοιχεία του πελάτη (στη customers) και μία για τις κρατήσεις του (στη bookings, φιλτραρισμένη με customer_id). Αν χρειάζονται πλήρεις λεπτομέρειες πτήσεων/ξενοδοχείων κ.λπ. για κάθε κράτηση, απαιτούνται επιπλέον \$lookup stages, αυξάνοντας την πολυπλοκότητα και τον χρόνο απόκρισης σε σύγκριση με την ενσωμάτωση.
- **β.2 (Πτήσεις βάσει κριτηρίων):** Όπως και στο (α.3), πολύ αποτελεσματικό.
- **β.3 (Πελάτες/συγκεκριμένη εταιρία ενοικίασης):** Απαιτεί query στη bookings, πιθανώς με \$unwind στο items, φιλτράρισμα για type: "car" και το company_name (που θα έπρεπε ίσως να αντιγραφεί στο items ή να απαιτεί \$lookup), και μετά \$group by customer_id. Λιγότερο άμεσο και δυνητικά πιο αργό.
- **β.4 (Εκδρομές/ξεναγό):** Αποτελεσματικό με query (find) στη συλλογή tours φιλτράροντας με guide_id.
- **β.5 Πελάτες/συγκεκριμένη εκδρομή:** Απαιτεί query στη bookings, φιλτράρισμα των items για το συγκεκριμένο tour_id, και εξαγωγή των μοναδικών customer_id. Απαιτεί σάρωση της bookings.

Αυτός ο σχεδιασμός διατηρεί τις συλλογές σχετικά "καθαρές" και στοχευμένες, κάνοντας τα queries που φιλτράρουν ή αθροίζουν δεδομένα εντός μίας συλλογής (π.χ. πτήσεις, ξενοδοχεία) πολύ γρήγορα. Οι σχέσεις υλοποιούνται κυρίως μέσω references (ObjectId) στην κεντρική συλλογή bookings.

- **3.2 Σχεδιασμός B: Βελτιστοποίηση για Ερωτήματα (β)**
 - *Φιλοσοφία:* Ο Σχεδιασμός B δίνει απόλυτη προτεραιότητα στην ταχύτητα ανάγνωσης για συγκεκριμένα, συχνά ερωτήματα, ειδικά αυτά που ζητούν όλες τις σχετικές πληροφορίες για μία κύρια οντότητα (π.χ., πελάτης, ξεναγός). Αυτό επιτυγχάνεται μέσω εκτεταμένης χρήσης ενσωμάτωσης (embedding). Σχετικά δεδομένα, όπως οι κρατήσεις ενός πελάτη ή οι ανατεθειμένες εκδρομές ενός ξεναγού, αποθηκεύονται απευθείας μέσα στο έγγραφο της κύριας οντότητας. Ο στόχος είναι η ανάκτηση όλων των απαραίτητων πληροφοριών με μία μόνο λειτουργία ανάγνωσης από τη βάση δεδομένων.
 - *Δομή Συλλογών (customers [με embedded bookings], flights, hotels, car_rentals, tour_guides [με assigned tours], tours [με participants]):*

1. customers: Ενσωματώνει τις κρατήσεις και τις εκδρομές που συμμετείχε.

```
{
  "_id": ObjectId("60f5a9f7e8b4a0b3e8f1b1b1"),
  "first_name": "Γιώργος",
  "last_name": "Παπαδόπουλος",
  "email": "mai24000@uom.edu.gr",
  "phone": "2311234567",
  "address": {
```

```

    "street": "Λεωφ. Αθηνών 10",
    "city": "Θεσσαλονίκη",
    "postal_code": "54242",
    "country": "Ελλάδα"
  },
  "registration_date": ISODate("2025-01-15T10:00:00Z"),
  "bookings": [ // Ενσωματωμένη λίστα κρατήσεων
    {
      "booking_id": ObjectId("60f5ab4ae8b4a0b3e8f1b1b7"), // Μοναδικό ID
      κράτησης
      "booking_date": ISODate("2025-01-15T15:30:00Z"),
      "total_price_eur": 1150.50,
      "status": "Confirmed",
      "flight": { // Ενσωματωμένες λεπτομέρειες πτήσης
        "flight_id": ObjectId("60f5aa3be8b4a0b3e8f1b1b2"), // Reference
        "flight_number": "A3 501",
        "airline": "Aegean Airlines",
        "departure_city": "Θεσσαλονίκη",
        "arrival_city": "Αθήνα",
        "departure_datetime": ISODate("2025-01-20T14:30:00Z"),
        "price_paid_eur": 80.50
      },
      "hotel": { // Ενσωματωμένες λεπτομέρειες ξενοδοχείου
        "hotel_id": ObjectId("60f5aa6ce8b4a0b3e8f1b1b3"), // Reference
        "hotel_name": "Hotel President",
        "city": "Αθήνα",
        "check_in_date": ISODate("2025-01-20T15:00:00Z"),
        "check_out_date": ISODate("2025-01-25T11:00:00Z"),
        "price_paid_eur": 900.00
      },
      "car_rental": { // Ενσωματωμένες λεπτομέρειες ενοικίασης
        "rental_id": ObjectId("60f5aa9be8b4a0b3e8f1b1b4"),
        "company_name": "Hertz",
        "pickup_location": "Ελ. Βενιζέλος (ΑΤΗ)",
        "pickup_datetime": ISODate("2025-01-20T10:00:00Z"),
        "dropoff_datetime": ISODate("2025-01-23T10:00:00Z"),
        "car_model": "BYD", // Προστέθηκε από το car_rentals doc
        "price_paid_eur": 135.00
      },
      "tour": { // Ενσωματωμένες λεπτομέρειες εκδρομής
        "tour_id": ObjectId("60f5aad5e8b4a0b3e8f1b1b5"), // Reference
        "tour_name": "Ημερήσια Εκδρομή Ακρόπολη, Πλάκα,
        Μοναστηράκι",
        "tour_date": ISODate("2025-01-24T09:00:00Z"),
        "price_paid_eur": 35.00
      }
    }
  ]
  // ... άλλες κρατήσεις του πελάτη ...

```



```
]
}
```

2. flights, hotels, car_rentals: Θα μπορούσαν να υπάρχουν ως ξεχωριστές συλλογές (όπως στο Σχεδιασμό Α) για γενικές αναζητήσεις (π.χ. όλες οι πτήσεις προς Χ), αλλά τα κρίσιμα δεδομένα για *συγκεκριμένες* κρατήσεις ενσωματώνονται στον πελάτη. Η συλλογή car_rentals θα περιέχει τουλάχιστον τη βασική πληροφορία για κάθε εταιρεία ενοικίασης.

```
{
  "_id": ObjectId("60f5aa3be8b4a0b3e8f1b1b2"),
  "airline": "Aegean Airlines",
  "flight_number": "A3 501",
  "departure": {
    "airport_code": "SKG",
    "city": "Θεσσαλονίκη",
    "country": "Ελλάδα",
    "datetime": ISODate("2025-01-20T14:30:00Z")
  },
  "arrival": {
    "airport_code": "ATH",
    "city": "Αθήνα",
    "country": "Ελλάδα",
    "datetime": ISODate("2025-01-20T15:15:00Z")
  },
  "price_eur": 80.50,
  "available_seats": 120
}
```

```
// hotels: Για γενικές αναζητήσεις ξενοδοχείων
{
  "_id": ObjectId("60f5aa6ce8b4a0b3e8f1b1b3"),
  "name": "Hotel President",
  "city": "Αθήνα",
  "country": "Ελλάδα",
  "stars": 5,
  "address": "Λεωφ. Κηφισίας 43",
  "amenities": ["Pool", "WiFi", "Restaurant", "Spa"],
  "price_per_night_eur": 180.00,
  "contact_email": "president@president.gr"
}
```

```
// car_rentals: Για γενικές αναζητήσεις ενοικιάσεων
{
  "_id": ObjectId("60f5aa9be8b4a0b3e8f1b1b4"),
  "company_name": "Hertz",
  "pickup_location": {
    "city": "Αθήνα",
```

```

        "point": "Ελ. Βενιζέλος (ΑΘΗ)"
    },
    "car_model": "BYD",
    "category": "EV",
    "price_per_day_eur": 40.00,
    "available_from": ISODate("2025-01-20T00:00:00Z"),
    "available_until": ISODate("2025-01-23T23:59:59Z")
    // Θα μπορούσαν να προστεθούν εδώ γενικές πληροφορίες στόλου
    // "fleet_categories": ["Mini", "Economy", "SUV", "EV"]
}

```

3. tour_guides: Ενσωματώνει τις εκδρομές που έχουν αναλάβει.

```

{
  "_id": ObjectId("60f5ab0fe8b4a0b3e8f1b1b6"),
  // Ας υποθέσουμε ότι το όνομα είναι Ελένη όπως στο παράδειγμα Β του
  prompt
  "first_name": "Γεωργία",
  "last_name": "Νικολάου",
  "languages": ["Ελληνικά", "Αγγλικά"],
  "specialization": ["Ιστορία", "Αρχαιολογία"],
  "contact_phone": "6999999999",
  // ... άλλες πληροφορίες οδηγού ...
  "assigned_tours": [ // Ενσωματωμένη λίστα IDs (ή και λεπτομερειών)
    εκδρομών
    {
      "tour_id": ObjectId("60f5aad5e8b4a0b3e8f1b1b5"),
      // Μπορούμε να προσθέσουμε και βασικά στοιχεία για ευκολία
      "tour_name": "Ημερήσια Εκδρομή Ακρόπολη, Πλάκα, Μοναστηράκι",
      "next_scheduled_date": ISODate("2025-01-24T09:00:00Z")
    }
    // ... άλλες ανατεθειμένες εκδρομές ...
  ]
}

```

4. tours: Ενσωματώνει τους πελάτες που έχουν συμμετάσχει (ή References).

```

{
  "_id": ObjectId("60f5aad5e8b4a0b3e8f1b1b5"),
  "name": "Ημερήσια Εκδρομή Ακρόπολη, Πλάκα, Μοναστηράκι",
  "destination_city": "Αθήνα",
  "duration_hours": 6,
  "description": "Εκδρομή στην Ακρόπολη, γεύμα στην Πλάκα, αγορές στο Μοναστηράκι",
  "price_eur": 35.00,
  "guide_id": ObjectId("60f5ab0fe8b4a0b3e8f1b1b6"), // Reference στον ξεναγό
  "scheduled_dates": [ // Οι προγραμματισμένες ημερομηνίες παραμένουν

```

```

        ISODate("2025-01-24T09:00:00Z"),
        ISODate("2025-01-25T09:00:00Z")
    ],
    // ... άλλες πληροφορίες εκδρομής ...
    "participants": [ // Λίστα συμμετεχόντων (μπορεί να είναι απλά IDs ή με
        λεπτομέρειες)
        {
            "customer_id": ObjectId("60f5a9f7e8b4a0b3e8f1b1b1"),
            "booking_ref_id": ObjectId("60f5ab4ae8b4a0b3e8f1b1b7"),
            "participation_date": ISODate("2025-01-24T09:00:00Z")
        }
        // ... άλλοι συμμετέχοντες για αυτή την εκδρομή/ημερομηνία ...
    ]
    // Εναλλακτικά, το participants array θα μπορούσε να είναι απλά IDs:
    // "participant_customer_ids": [ObjectId("60f5a9f7e8b4a0b3e8f1b1b1"), ...]
}

```

5. **bookings**: Αυτή η συλλογή μπορεί να είναι πιο λιτή τώρα, ή ακόμα και να *παραλειφθεί* αν όλες οι πληροφορίες κρατήσεων βρίσκονται μόνο μέσα στους πελάτες. Αν διατηρηθεί, θα χρησίμευε ίσως για global queries όπως "όλες οι κρατήσεις μεταξύ ημερομηνιών X και Ψ" (α.2), αλλά ο σχεδιασμός αυτός δεν δίνει προτεραιότητα σε αυτό. Ας υποθέσουμε ότι την παραλείπουμε για να τονίσουμε την ενσωμάτωση.

Τεκμηρίωση Επιλογών (Σχεδιασμός Β):

Για Ερωτήματα (β):

- **β.1 (Κρατήσεις συγκεκριμένου πελάτη;)**: Εξαιρετικά γρήγορο. Μία μόνο ανάγνωση του εγγράφου του πελάτη από τη συλλογή customers δίνει όλες τις κρατήσεις και τις λεπτομέρειές τους (λόγω embedding). `db.customers.findOne({ _id: ObjectId("...") })`
- **β.2 (Πτήσεις/προορισμό/προέλευση/ημερομηνία;)**: Παραμένει αποδοτικό, χρησιμοποιώντας την ξεχωριστή συλλογή flights (ίδιο query με το Σχέδιο Α).
- **β.3 (Πελάτες/συγκεκριμένη εταιρία ενοικίασης;)**: Εφικτό αλλά πιθανώς λιγότερο αποδοτικό από το Σχέδιο Α αν η βάση είναι τεράστια. Απαιτεί query στη συλλογή customers που ψάχνει *μέσα* στο ενσωματωμένο bookings array. Χρειάζεται οπωσδήποτε index στο bookings.car_rental.company_name. `db.customers.find({"bookings.car_rental.company_name": "Hertz" })`
- **β.4 (Εκδρομές/ξεναγό;)**: Πολύ γρήγορο. Ανάκτηση του εγγράφου του ξεναγού από τη tour_guides δίνει άμεσα τις εκδρομές που έχει αναλάβει (μέσω του assigned_tours). `db.tour_guides.findOne({ _id: ObjectId("...") })`
- **β.5 (Πελάτες/συγκεκριμένη εκδρομή;)**: Πολύ γρήγορο. Ανάκτηση του εγγράφου της εκδρομής από τη tours δίνει άμεσα τη λίστα των συμμετεχόντων (μέσω του participants). `db.tours.findOne({ _id: ObjectId("...") })`

Για Ερωτήματα (α):

- **α.1 (Πόσες κρατήσεις/πελάτη):** Αποτελεσματικό μέσω aggregation στη customers, χρησιμοποιώντας \$project για να πάρουμε το μέγεθος (\$size) του πίνακα bookings. Διαφορετική προσέγγιση από τον Σχεδιασμό A, αλλά γρήγορη.
- **α.2 (Κρατήσεις/χρονικό διάστημα):** Πιθανώς πολύ πιο αργό και πολύπλοκο. Απαιτεί σάρωση όλων των εγγράφων στη συλλογή customers και φιλτράρισμα μέσα στον ενσωματωμένο πίνακα bookings για κάθε έγγραφο. Δεν είναι η βέλτιστη προσέγγιση για τέτοιου είδους range queries.
- **α.3 (Πτήσεις βάσει κριτηρίων):** Πολύ αποτελεσματικό (ίδιο με Σχεδιασμό A).
- **α.4 (Ξενοδοχεία/εύρος τιμών):** Πολύ αποτελεσματικό (ίδιο με Σχεδιασμό A).
- **α.5 (Εκδρομές/πελάτη):** Εξαιρετικά γρήγορο (ίδιο αποτέλεσμα με β.1 ουσιαστικά, απλά φιλτράρουμε τοπικά τον πίνακα bookings για items τύπου tour μετά την ανάγνωση του εγγράφου του πελάτη).

Αυτός ο σχεδιασμός μειώνει δραστικά την ανάγκη για \$lookup (joins) για τα συγκεκριμένα ερωτήματα (β), κάνοντας τις αντίστοιχες αναγνώσεις πολύ γρήγορες. Το τίμημα είναι η πιθανή επανάληψη δεδομένων (data duplication - π.χ. όνομα ξενοδοχείου σε κάθε κράτηση πελάτη) και το ότι ορισμένα άλλα queries (όπως το β.3) μπορεί να απαιτούν σάρωση ενσωματωμένων πεδίων, η οποία, αν και εφικτή με indexes, μπορεί να είναι πιο αργή σε πολύ μεγάλα datasets σε σύγκριση με το query σε μια πιο "επίπεδη" συλλογή όπως οι bookings του Σχεδιασμού A. Επίσης, οι ενημερώσεις σε δεδομένα που επαναλαμβάνονται (π.χ. αλλαγή ονόματος ξενοδοχείου) απαιτούν ενημέρωση σε πολλαπλά έγγραφα.

4. Συμπεράσματα & Προτάσεις

Η ανάλυση των δύο εναλλακτικών σχεδιασμών βάσης δεδομένων MongoDB για το ταξιδιωτικό γραφείο καταδεικνύει τη θεμελιώδη σημασία της κατανόησης των κυρίαρχων προτύπων πρόσβασης στα δεδομένα (query patterns) κατά τη φάση του σχεδιασμού. Δεν υπάρχει ένας μοναδικός "σωστός" τρόπος μοντελοποίησης, αλλά μια σειρά από trade-offs που πρέπει να σταθμιστούν.

- **Ο Σχεδιασμός A**, με την έμφαση στις αναφορές, προσφέρει μεγαλύτερη ευελιξία για ad-hoc ερωτήματα, μειώνει την επανάληψη δεδομένων και απλοποιεί τις ενημερώσεις σε οντότητες όπως τα ξενοδοχεία ή οι πτήσεις. Είναι ιδανικός εάν η εφαρμογή εκτελεί συχνά συγκεντρωτικές αναλύσεις, στατιστικές αναφορές (ερωτήματα τύπου α) ή απαιτεί συνεπή δεδομένα σε όλο το σύστημα. Το κόστος είναι η πιθανή ανάγκη για πολλαπλές αναγνώσεις ή τη χρήση \$lookup για τη σύνθεση ολοκληρωμένων προβολών (π.χ., όλες οι λεπτομέρειες μιας κράτησης).
- **Ο Σχεδιασμός B**, με την εκτεταμένη χρήση ενσωμάτωσης, παρέχει εξαιρετική απόδοση ανάγνωσης για τα συγκεκριμένα ερωτήματα για τα οποία βελτιστοποιήθηκε (ερωτήματα τύπου β), όπως η ανάκτηση όλων των κρατήσεων ενός πελάτη με μία μόνο πρόσβαση στη βάση. Αυτό μπορεί να είναι κρίσιμο για use cases που απαιτούν γρήγορη φόρτωση τέτοιων προβολών. Τα μειονεκτήματα περιλαμβάνουν την πιθανή επανάληψη δεδομένων, τα μεγαλύτερα έγγραφα και την αυξημένη πολυπλοκότητα κατά την ενημέρωση δεδομένων που υπάρχουν σε πολλαπλά σημεία (π.χ., αλλαγή ονόματος ξενοδοχείου).

Πρόταση:

Η επιλογή μεταξύ των δύο σχεδιασμών θα πρέπει να βασιστεί στα πιο συχνά και κρίσιμα για την απόδοση ερωτήματα της εφαρμογής:

- Εάν οι **συγκεντρωτικές αναφορές, οι στατιστικές και οι ευρείες αναζητήσεις** (ερωτήματα α) είναι πρωταρχικής σημασίας, προτείνεται ο **Σχεδιασμός Α**.
- Εάν η **ταχύτατη ανάκτηση του πλήρους προφίλ μιας οντότητας** (π.χ., όλες οι κρατήσεις του πελάτη Χ, οι εκδρομές του ξεναγού Ψ - ερωτήματα β) είναι η βασική απαίτηση, προτείνεται ο **Σχεδιασμός Β**.

Είναι επίσης δυνατόν να εφαρμοστεί μια **υβριδική προσέγγιση**, ξεκινώντας ίσως με τον Σχεδιασμό Α και εφαρμόζοντας επιλεκτική ενσωμάτωση (denormalization) για συγκεκριμένα πεδία ή σχέσεις που απαιτούν βελτιστοποίηση στην ανάγνωση, παρακολουθώντας πάντα τις επιπτώσεις στην απόδοση και τη συντηρησιμότητα.

Ερώτημα Β

1. Εμφανίστε το πλήθος των εγγράφων που υπάρχουν στη συλλογή movies (ταινίες).

```
db.movies.countDocuments()
```

2. Εμφανίστε μόνο τον τίτλο, το έτος, τους σκηνοθέτες και τα είδη για τις 20 πιο πρόσφατες ταινίες, ταξινομημένες κατά έτος και αλφαβητική σειρά.

```
db.movies.find(  
  { directors: { $exists: true } },  
  { title: 1, year: 1, directors: 1, genres: 1, _id: 0 } )  
  .sort({ year: -1, title: 1 }).limit(20)
```

3. Εμφανίστε τον τίτλο και το έτος για τις ταινίες για τις ταινίες που κυκλοφόρησαν μετά το 1970 και έχουν rated "NOT RATED".

```
db.movies.find(  
  { year: { $gt: 1970 }, rated: "NOT RATED" },  
  { title: 1, year: 1, _id: 0 } )
```

4. Εμφανίστε τα στοιχεία των ταινιών που έχουν ως είδος "Comedy" ή "Romance".

```
db.movies.find(  
  genres: { $in: ["Comedy", "Romance"] } )
```

5. Εμφανίστε τα στοιχεία των ταινιών στις οποίες παίζει ο "Tom Hanks" και η "Meg Ryan".

```
db.movies.find(  
  cast: { $all: ["Tom Hanks", "Meg Ryan"] } )
```

6. Εμφανίστε τα στοιχεία των ταινιών στις οποίες παίζει ο “Hugo Weaving” αλλά όχι ο “Keanu Reeves”.

```
db.movies.find({
  cast: { $in: ["Hugo Weaving"], $nin: ["Keanu Reeves"] })})
```

7. Εμφανίστε το μέσο πλήθος σχολίων (num_mflix_comments) που έχουν οι ταινίες.

```
db.movies.aggregate([
  { $group: { _id: null, avgComments: { $avg: "$num_mflix_comments" } } })})
```

8. Εμφανίστε τη μέγιστη και την ελάχιστη διάρκεια των ταινιών παραγωγής “USA”.

```
db.movies.aggregate([
  { $match: { countries: "USA" } },
  { $group: {
    _id: null,
    maxRuntime: { $max: "$runtime" },
    minRuntime: { $min: "$runtime" } } })})
```

9. Εμφανίστε το πλήθος των ταινιών ανά χαρακτηρισμό ταινίας (rated) και έτος.

```
db.movies.aggregate([
  { $group: {
    _id: { rated: "$rated", year: "$year" },
    count: { $sum: 1 } } },
  { $sort: { "_id.year": 1, "_id.rated": 1 } })})
```

10. Εμφανίστε το πλήθος και το συνολικό αριθμό σχολίων (num_mflix_comments) για τις ταινίες που έχουν διάρκεια άνω των 100 λεπτών ανά έτος.

```
db.movies.aggregate([
  { $match: { runtime: { $gt: 100 } } },
  { $group: {
    _id: "$year",
    count: { $sum: 1 },
    totalComments: { $sum: "$num_mflix_comments" }
  } },
  { $sort: { _id: 1 } })})
```

11. Εμφανίστε το πλήθος των ταινιών που έχουν τουλάχιστον 5 υποψηφιότητες για βραβεία ανά έτος, για τα έτη που έχουν τουλάχιστον 3 τέτοιες ταινίες.

```
db.movies.aggregate([
  { $match: { "awards.nominations": { $gte: 5 } } },
  { $group: {
    _id: "$year",
    count: { $sum: 1 } } },
  { $sort: { _id: 1 } })})
```

```
{ $match: { count: { $gte: 3 } } },  
{ $sort: { _id: 1 } } ]])
```

12. Εμφανίστε το πλήθος και τη λίστα των τίτλων των ταινιών ανά είδος για τις ταινίες στις οποίες έχει παίξει ο "Keanu Reeves".

```
db.movies.aggregate([  
  { $match: { cast: "Keanu Reeves" } },  
  { $unwind: "$genres" },  
  { $group: {  
    _id: "$genres",  
    count: { $sum: 1 },  
    titles: { $push: "$title" } } ]])
```

13. Εμφανίστε το πλήθος και τη λίστα των ταινιών ανά χώρα και είδος, για τα ζεύγη που έχουν λιγότερες από 20 ταινίες.

```
db.movies.aggregate([  
  { $unwind: "$countries" },  
  { $unwind: "$genres" },  
  { $group: {  
    _id: { country: "$countries", genre: "$genres" },  
    count: { $sum: 1 },  
    titles: { $push: "$title" } } },  
  { $match: { count: { $lt: 20 } } } ]])
```

14. Εμφανίστε τον ηθοποιό που έχει παίξει στις περισσότερες ταινίες.

```
db.movies.aggregate([  
  { $unwind: "$cast" },  
  { $group: {  
    _id: "$cast",  
    count: { $sum: 1 } } },  
  { $sort: { count: -1 } },  
  { $limit: 1 } ]])
```

15. Εμφανίστε τις ταινίες για τις οποίες έχει κάνει σχόλια ο χρήστης με όνομα "Jon Snow" (συλλογή comments και movies).

```
db.comments.aggregate([  
  { $match: { name: "Jon Snow" } },  
  { $lookup: {  
    from: "movies",  
    localField: "movie_id",  
    foreignField: "_id",  
    as: "movie"  
  } },  
  { $unwind: "$movie" },
```

```
{ $replaceRoot: { newRoot: "$movie" } })
```