



## **CIT6234 - ADVANCED DATABASE**

### **ASSIGNMENT 2 (30%) GROUP 4**

#### **CS3: Data warehouse for AirAsia flight ticket booking system**

Design a data warehouse to keep track of flight ticket booking system for AirAsia (<https://www.airasia.com/flight/>). This include information such as destination from, destination to, date, time, promotion, and so on.

**Prepared by:**

<b>Name</b>	<b>Student ID</b>	<b>Student Email</b>
Nur Fatin Nabilah Binti Md.Irzan	1221302587	1221302587@student.mmu.edu.my
Nur Alia Shazwani Binti Mohd Nazri	1231302985	1231302985@student.mmu.edu.my
Haizatul Nazirah Nizam Binti Hairunizam	1231303504	1231303504@student.mmu.edu.my
Wong Jin Yin	1211100975	1211100975@student.mmu.edu.my

# Table of Contents

<b>CS3: Data warehouse for AirAsia flight ticket booking system.....</b>	<b>1</b>
<b>1.0 Data Modelling .....</b>	<b>3</b>
<b>2.0 Data Dictionary .....</b>	<b>5</b>
<b>3.0 Database and Respective Collections.....</b>	<b>7</b>
<b>4.0 Documents in Collections.....</b>	<b>11</b>
<b>5.0 Complex Query .....</b>	<b>27</b>
<b>5.1 Logical Operation.....</b>	<b>27</b>
<b>5.2 Comparison Operator.....</b>	<b>28</b>
<b>5.3 Aggregate Function.....</b>	<b>29</b>
<b>5.4 Project Command .....</b>	<b>30</b>
<b>5.5 Size Clause .....</b>	<b>31</b>
<b>6.0 NoSQL Command to Update Particular Record(s) .....</b>	<b>33</b>
<b>8.0 Two NoSQL Command Not Covered in Lecture.....</b>	<b>36</b>
<b>8.1 Function findOneAndReplace() .....</b>	<b>36</b>
<b>8.2 Function distinct() .....</b>	<b>37</b>
<b>9.0 Part B: Essay on The Impact of Big Data in Airlines Industry .....</b>	<b>38</b>

## 1.0 Data Modelling

### Embedded Document

Sales {

sales\_id: "AB123456",

Customer: {

cust\_id: 10077,

cust\_name: "Emily Smith",

cust\_age: 25,

cust\_passport: "AB123456",

cust\_gender: "FEMALE",

cust\_contactnum: "+15551234567",

cust\_email: "emilysm@gmail.com"

},

Location: {

loc\_id: "KUL",

loc\_country: "Malaysia",

loc\_state: "Kuala Lumpur",

loc\_city: "Kuala Lumpur"

},

Time: {

time\_id: 1679802000,

time\_quarter: 2,

time\_month: "April",

time\_day: "Saturday",

time\_date: "2024-04-20"

},

Company: {

company\_id: "AXM",

```
    company_name: "AirAsia Malaysia",
    company_loc: "Kuala Lumpur",
    company_email: "maa_groupdesk@airasia.com",
    company_contactnum: "+60378411818",
    company_rating: 3.1
  },
```

```
Promotion: {
  promo_code: "WELCOME10",
  promo_desc: "GET 10% OFF",
  promo_disc: 10.0,
  promo_start: "2024-04-01",
  promo_end: "2024-07-31"
},
```

```
Payment: {
  payment_id: 1234567890,
  payment_method: "Credit Card",
},
```

```
Flight: {
  flight_no: "AK123",
  flight_origin: "KUL",
  flight_destination: "DPS",
  flight_departdate: "2024-07-15",
  flight_departtime: "19:45:00",
  flight_arrivaldate: "2024-07-15",
  flight_arrivaltime: "22:45:00",
},
```

```
Sale_total: 329.00
}
```

## 2.0 Data Dictionary

Collections Name	Object Name	Fields	Data Type	Unique
Sales	_id		ObjectID	True
	Customer	7	Object	
	id		ObjectID	
	name		String	
	age		String	
	passport		String	
	gender		String	
	contactnum		String	
	email		String	
	Location	4	Object	
	id		ObjectID	
	country		String	
	state		String	
	city		String	
	Time	5	Object	
	id		ObjectID	
	quarter		String	
	month		String	
	day		String	
	date		Date	
	Company	6	Object	
	id		ObjectID	
	name		String	
	loc		String	
	email		String	
	contactnum		String	
	rating		String	
	Promotion	5	Object	

	code		String	
	desc		String	
	disc		String	
	start		Date	
	end		Date	
	Payment	2	Object	
	id		ObjectID	
	method		String	
	Flight	7	Object	
	no		String	
	origin		String	
	destination		String	
	departdate		Date	
	departtime		String	
	arrivaldate		Date	
	arrivaltime		String	
	Sale_total		Double	

### 3.0 Database and Respective Collections

Implement the data warehouse by modelling the database using a document-based NoSQL.

- i. Create the database and respective collections using MongoDB

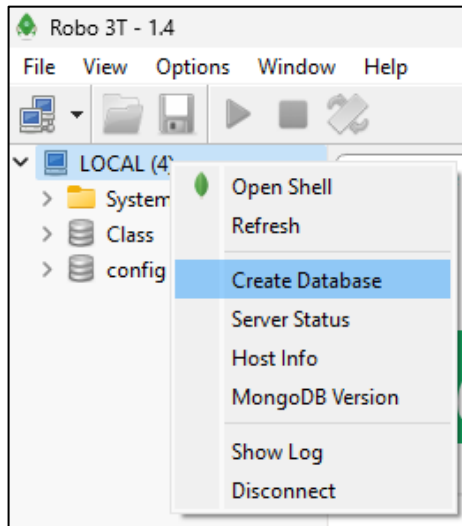


Figure 3.1 Creating new database by selecting “Create Database”

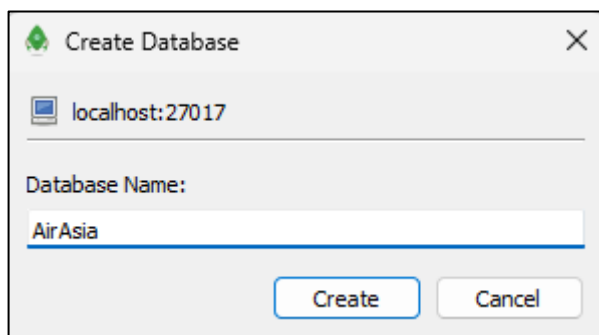


Figure 3.2 Naming the database as “AirAsia”

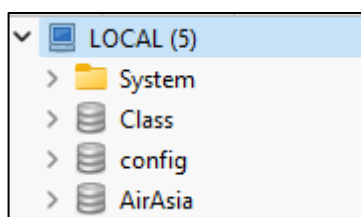


Figure 3.3 Database created

## ii. Creating Collections using MongoDB

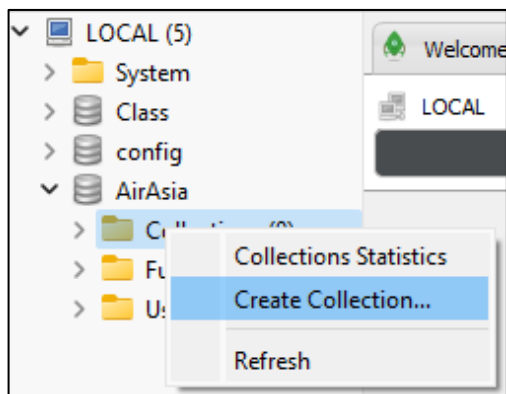


Figure 3.4 Creating Collections

- Customer

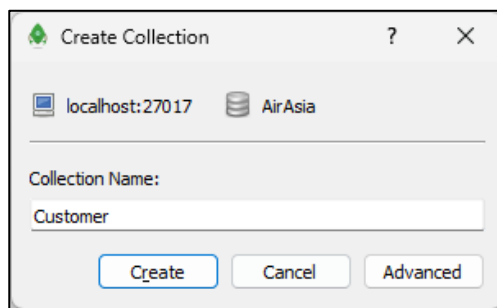


Figure 3.4.1 Collection Name "Customer"

- Location

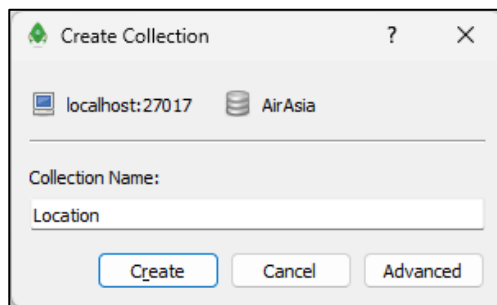


Figure 3.4.1 Collection Name "Location"

- Company

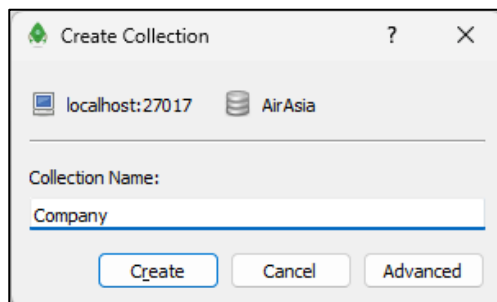


Figure 3.4.1 Collection Name "Company"



- Promotion

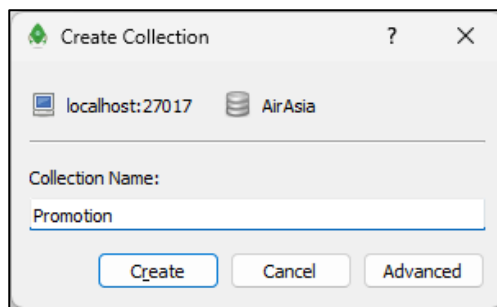


Figure 3.4.1 Collection Name "Promotion"

- Payment

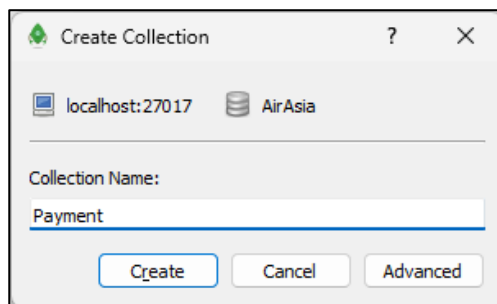


Figure 3.4.1 Collection Name "Payment"

- Flight

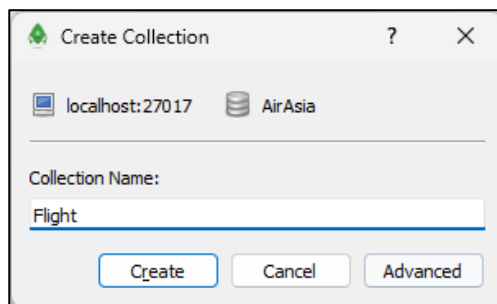


Figure 3.4.1 Collection Name "Flight"

- Time

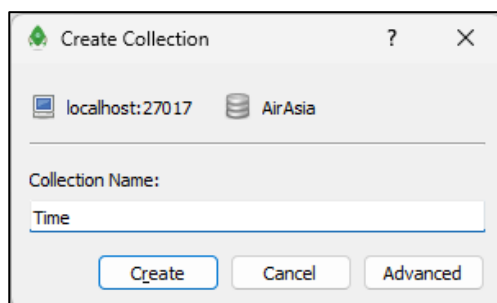


Figure 3.4.1 Collection Name "Time"

- Sales

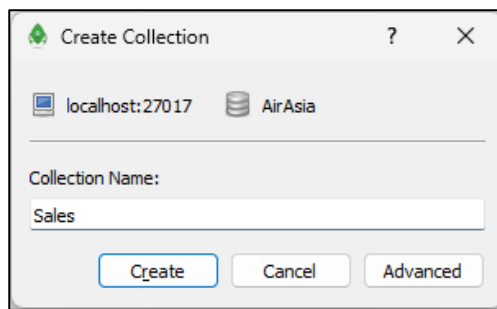


Figure 3.4.1 Collection Name “Sales”

## 4.0 Documents in Collections

Enter sample data into the collections using MongoDB.

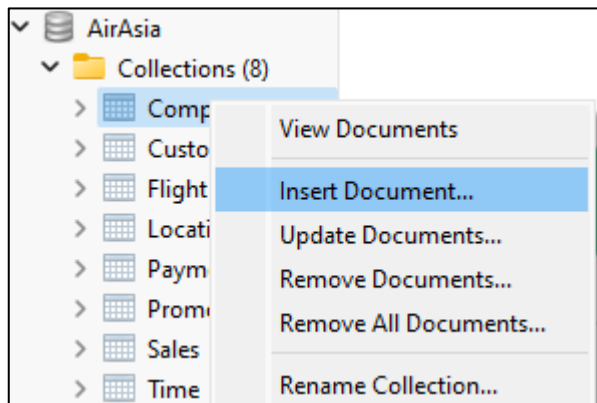


Figure 4.0.1 Insert Document into the Collections

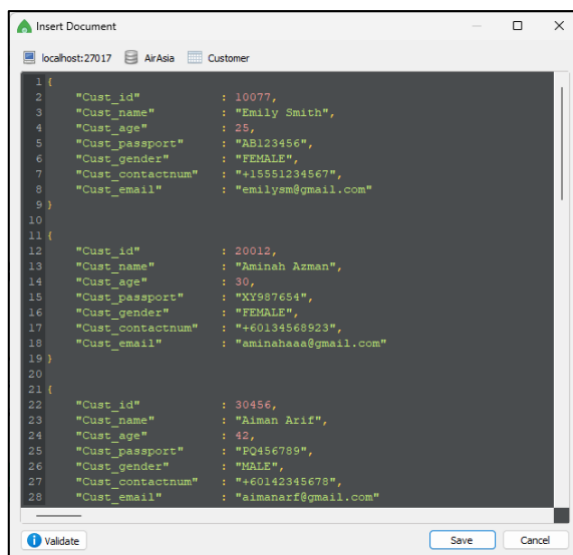


Figure 4.0.2 Insert 10 sample documents into the Collections

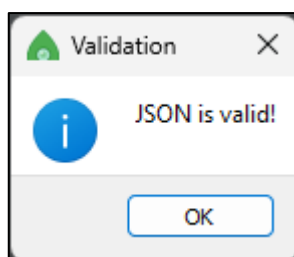


Figure 4.0.3 Validate JSON for any error in insert

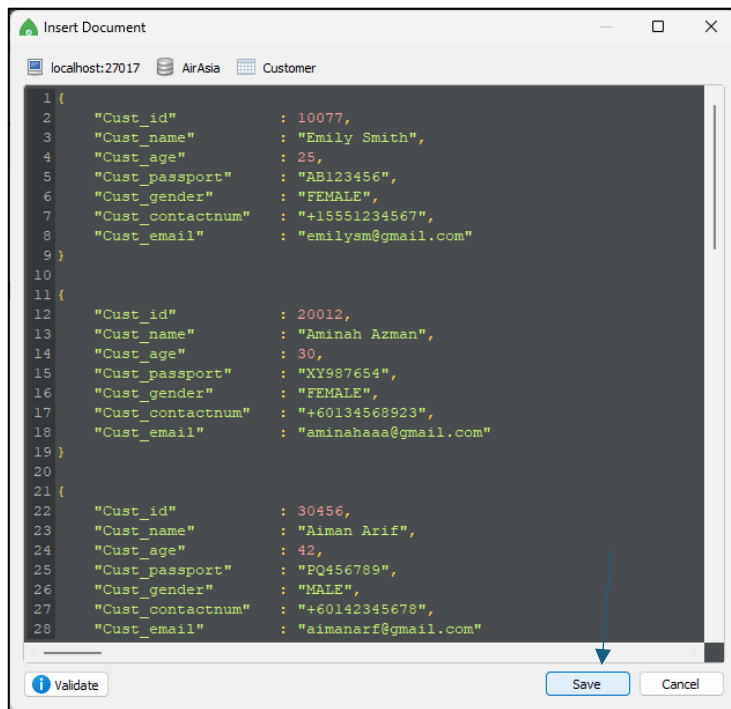


Figure 4.0.4 Save the document

- Customer

<pre>{ "Cust_id"      : 10077,   "Cust_name"    : "Emily Smith",   "Cust_age"     : 25,   "Cust_passport" : "AB123456",   "Cust_gender"  : "FEMALE",   "Cust_contactnum" : "+15551234567",   "Cust_email"   : "emilysm@gmail.com" } {   "Cust_id"      : 20012,   "Cust_name"    : "Aminah Azman",   "Cust_age"     : 30,   "Cust_passport" : "XY987654",   "Cust_gender"  : "FEMALE",   "Cust_contactnum" : "+60134568923",   "Cust_email"   : "aminahaaa@gmail.com" } {   "Cust_id"      : 30456,   "Cust_name"    : "Aiman Arif",   "Cust_age"     : 42,   "Cust_passport" : "PQ456789",   "Cust_gender"  : "MALE",   "Cust_contactnum" : "+60142345678",   "Cust_email"   : "aimanarf@gmail.com" } {   "Cust_id"      : 70897,   "Cust_name"    : "Lee Soon Ye",   "Cust_age"     : 28,   "Cust_passport" : "LJ391728",   "Cust_gender"  : "MALE",   "Cust_contactnum" : "+442012322678",   "Cust_email"   : "soonye@gmail.com" } {   "Cust_id"      : 11384,   "Cust_name"    : "Wan Seri",   "Cust_age"     : 53,</pre>	<pre>{   "Cust_id"      : 40123,   "Cust_name"    : "Jennifer Lee",   "Cust_age"     : 19,   "Cust_passport" : "CD789012",   "Cust_gender"  : "FEMALE",   "Cust_contactnum" : "+60137777456",   "Cust_email"   : "jenniferlee@yahoo.com" } {   "Cust_id"      : 50234,   "Cust_name"    : "Daniel Iman",   "Cust_age"     : 36,   "Cust_passport" : "EF234567",   "Cust_gender"  : "MALE",   "Cust_contactnum" : "+60115968275",   "Cust_email"   : "daniell@gmail.com" } {   "Cust_id"      : 60567,   "Cust_name"    : "Kumar Singh",   "Cust_age"     : 50,   "Cust_passport" : "GH345678",   "Cust_gender"  : "MALE",   "Cust_contactnum" : "+60122323979",   "Cust_email"   : "kumarsk@yahoo.com" } {   "Cust_id"      : 77239,   "Cust_name"    : "Ahmad Ali",</pre>
---	--

<pre> "Cust_passport" : "AP493789", "Cust_gender"   : "FEMALE", "Cust_contactnum" : "+60138955789", "Cust_email"    : "wanseri@gmail.com" } {   "Cust_id"      : 90567,   "Cust_name"    : "Sophia Adam",   "Cust_age"     : 22,   "Cust_passport" : "ST398734",   "Cust_gender"  : "FEMALE",   "Cust_contactnum" : "+6121345678",   "Cust_email"   : "sophiadm@gmail.com" } </pre>	<pre> "Cust_age"      : 44, "Cust_passport" : "ND827631", "Cust_gender"   : "MALE", "Cust_contactnum" : "+60145955657", "Cust_email"    : "ahmadali@gmail.com" } </pre>
---	---

Table 4.1 10 sample documents for Customer

Customer 0.007 sec.							
_id	Cust_id	Cust_name	Cust_age	Cust_passport	Cust_gender	Cust_contactnum	Cust_email
1	ObjectId("6... 10077	Emily Smith	25	AB123456	FEMALE	+15551234567	emilysm@gmail.com
2	ObjectId("6... 20012	Aminah Azman	30	XY987654	FEMALE	+60134568923	aminahaaa@gmail.com
3	ObjectId("6... 30456	Aiman Arif	42	PQ456789	MALE	+60142345678	aimanarf@gmail.com
4	ObjectId("6... 40123	Jennifer Lee	19	CD789012	FEMALE	+60137777456	jenniferlee@yahoo.com
5	ObjectId("6... 50234	Daniel Iman	36	EF234567	MALE	+60115968275	daniell@gmail.com
6	ObjectId("6... 60567	Kumar Singh	50	GH345678	MALE	+60122323979	kumarsk@yahoo.com
7	ObjectId("6... 70897	Lee Soon Ye	28	LJ391728	MALE	+442012322678	soonye@gmail.com
8	ObjectId("6... 77239	Ahmad Ali	44	ND827631	MALE	+60145955657	ahmadali@gmail.com
9	ObjectId("6... 90567	Sophia Adam	22	ST398734	FEMALE	+6121345678	sophiadm@gmail.com
10	ObjectId("6... 11384	Wan Seri	53	AP493789	FEMALE	+60138955789	wanseri@gmail.com

Figure 4.1 Table view of the 10 sample documents for Customer

- Location

<pre> { "Loc_id"      : "BNE",   "Loc_country" : "Australia",   "Loc_state"   : "Brisbane",   "Loc_city"    : "Brisbane" } { "Loc_id"      : "KUL",   "Loc_country" : "Malaysia",   "Loc_state"   : "Kuala Lumpur",   "Loc_city"    : "Kuala Lumpur" } { "Loc_id"      : "DPS",   "Loc_country" : "Indonesia",   "Loc_state"   : "Denpasar",   "Loc_city"    : "Bali" } {   "Loc_id"      : "MNL",   "Loc_country" : "Philippines",   "Loc_state"   : "Manila",   "Loc_city"    : "Manila" } {   "Loc_id"      : "ICN",   "Loc_country" : "South Korea",   "Loc_state"   : "Seoul",   "Loc_city"    : "Incheon" } </pre>	<pre> { "Loc_id"      : "SYD",   "Loc_country" : "Australia",   "Loc_state"   : "Sydney",   "Loc_city"    : "Sydney" } { "Loc_id"      : "KIX",   "Loc_country" : "Japan",   "Loc_state"   : "Osaka",   "Loc_city"    : "Osaka" } { "Loc_id"      : "TPE",   "Loc_country" : "Taiwan",   "Loc_state"   : "Taipei",   "Loc_city"    : "Taipei" } { "Loc_id"      : "SIN",   "Loc_country" : "Singapore",   "Loc_state"   : "Singapore",   "Loc_city"    : "Singapore" } {   "Loc_id"      : "BWN",   "Loc_country" : "Brunei",   "Loc_state"   : "Brunei",   "Loc_city"    : "Bandar Seri Begawan" } </pre>
--	--

Table 4.2 10 sample documents for Location

Location 0.005 sec.					
	_id	Loc_id	Loc_country	Loc_state	Loc_city
1	ObjectId("6...	BNE	Australia	Brisbane	Brisbane
2	ObjectId("6...	KUL	Malaysia	Kuala Lumpur	Kuala Lumpur
3	ObjectId("6...	DPS	Indonesia	Denpasar	Bali
4	ObjectId("6...	MNL	Philippines	Manila	Manila
5	ObjectId("6...	SYD	Australia	Sydney	Sydney
6	ObjectId("6...	KIX	Japan	Osaka	Osaka
7	ObjectId("6...	TPE	Taiwan	Taipei	Taipei
8	ObjectId("6...	SIN	Singapore	Singapore	Singapore
9	ObjectId("6...	ICN	South Korea	Seoul	Incheon
10	ObjectId("6...	BWN	Brunei	Brunei	Bandar Seri Begawan

Figure 4.2 Table view of the 10 sample documents for Location

- Company

<pre> {  "Company_id"      : "AXM",    "Company_name"    : "AirAsia",    "Company_loc"     : "Kuala Lumpur",    "Company_email": "maa_groupdesk@airasia.com",    "Company_contactnum": "+60378411818",    "Company_rating"  : 3.1 } {    "Company_id"      : "KTC",    "Company_name"    : "AirAsia Cambodia",    "Company_loc"     : "Phnom Penh",    "Company_email"   : "taa_pnhgrp@airasia.com",    "Company_contactnum": "+855236329979",    "Company_rating"  : 3.1 } {    "Company_id"      : "IAD",    "Company_name"    : "AirAsia India",    "Company_loc"     : "Chennai",    "Company_email"   : "in_groupdesk@airasia.com",    "Company_contactnum": "+918048101460",    "Company_rating"  : 2.5 } {    "Company_id"      : "WAJ",    "Company_name"    : "AirAsia Japan",    "Company_loc"     : "Nagoya",    "Company_email"   : "japan_groupdesk@airasia.com",    "Company_contactnum": "+815068648183",    "Company_rating"  : 3.1 } {    "Company_id"      : "XAX",    "Company_name"    : "AirAsia X",    "Company_loc"     : "Australia", </pre>	<pre> {    "Company_id"      : "AIQ",    "Company_name"    : "AirAsia Thai",    "Company_loc"     : "Bangkok",    "Company_email"   : "taa_groupdesk@airasia.com",    "Company_contactnum": "+6620297862",    "Company_rating"  : 3.1 } {    "Company_id"      : "TAX",    "Company_name"    : "AirAsia Thai X",    "Company_loc"     : "Bangkok",    "Company_email"   : "taax_groupdesk@airasia.com",    "Company_contactnum": "+6625159888",    "Company_rating"  : 3.5 } {    "Company_id"      : "EZD",    "Company_name"    : "AirAsia Zest",    "Company_loc"     : "Manila",    "Company_email"   : "aaz_groupdesk@airasia.com",    "Company_contactnum": "+60386600008",    "Company_rating"  : 3.1 } {    "Company_id"      : "APG", </pre>
--	---



<pre> "Company_email" : "aax_groupdesk@airasia.com", "Company_contactnum": "+61238138388", "Company_rating" : 3.5 } { "Company_id" : "AWQ", "Company_name" : "AirAsia Indonesia", "Company_loc" : "Jakarta", "Company_email" : "iaa_groupdesk@airasia.com", "Company_contactnum": "+622129850850", "Company_rating" : 3.5 } </pre>	<pre> "Company_name" : "AirAsia Philippines", "Company_loc" : "Manila", "Company_email" : "paa_groupdesk@airasia.com", "Company_contactnum": "+60263247715", "Company_rating" : 3.1 } </pre>
--	--

Table 4.3 10 sample documents for Company

Company 0.006 sec.							
	_id	Company_id	Company_name	Company_loc	Company_email	Company_contactnum	Company_rating
1	ObjectId("6...	AXM	AirAsia	Kuala Lumpur	maa_groupdesk@airasia.com	+60378411818	3.1
2	ObjectId("6...	KTC	AirAsia Cambodia	Phnom Penh	taa_pnhgrp@airasia.com	+855236329979	3.1
3	ObjectId("6...	IAD	AirAsia India	Chennai	in_groupdesk@airasia.com	+918048101460	2.5
4	ObjectId("6...	WAJ	AirAsia Japan	Nagoya	japan_groupdesk@airasia.com	+815068648183	3.1
5	ObjectId("6...	XAX	AirAsia X	Australia	aax_groupdesk@airasia.com	+61238138388	3.5
6	ObjectId("6...	AWQ	AirAsia Indonesia	Jakarta	iaa_groupdesk@airasia.com	+622129850850	3.5
7	ObjectId("6...	EZD	AirAsia Zest	Manila	aaz_groupdesk@airasia.com	+60386600008	3.1
8	ObjectId("6...	APG	AirAsia Philippines	Manila	paa_groupdesk@airasia.com	+60263247715	3.1
9	ObjectId("6...	AIQ	AirAsia Thai	Bangkok	taa_groupdesk@airasia.com	+6620297862	3.1
10	ObjectId("6...	TAX	AirAsia Thai X	Bangkok	taax_groupdesk@airasia.com	+6625159888	3.5

Figure 4.3 Table view of the 10 sample documents for Location

- Promotion

<pre>{   "Promo_code"      : "WELCOME10",   "Promo_desc"      : "Get 10% OFF",   "Promo_disc"      : 10.0,   "Promo_start"     : "2024-04-01",   "Promo_end"       : "2024-07-31" }</pre>	<pre>{   "Promo_code"      : "BIGPAY5",   "Promo_desc"      : "Get 5% OFF",   "Promo_disc"      : 5.00,   "Promo_start"     : "2024-06-01",   "Promo_end"       : "2024-07-31" }</pre>
<pre>{   "Promo_code"      : "WELCOMEAPP",   "Promo_desc"      : "Get 20% OFF",   "Promo_disc"      : 20.0,   "Promo_start"     : "2024-01-01",   "Promo_end"       : "2024-09-01" }</pre>	<pre>{   "Promo_code"      : "HOLIDAY",   "Promo_desc"      : "Get 60% OFF",   "Promo_disc"      : 60.0,   "Promo_start"     : "2024-03-01",   "Promo_end"       : "2024-05-31" }</pre>
<pre>{   "Promo_code"      : "LOWFARE",   "Promo_desc"      : "Get 5% OFF",   "Promo_disc"      : 5.00,   "Promo_start"     : "2024-03-01",   "Promo_end"       : "2024-06-29" }</pre>	<pre>{   "Promo_code"      : "WELCOME20",   "Promo_desc"      : "Get 20% OFF",   "Promo_disc"      : 20.0,   "Promo_start"     : "2024-05-01",   "Promo_end"       : "2024-07-30" }</pre>
<pre>{   "Promo_code"      : "MAYDEAL",   "Promo_desc"      : "Get 30% OFF",   "Promo_disc"      : 30.0,   "Promo_start"     : "2024-03-01",   "Promo_end"       : "2024-05-31" }</pre>	<pre>{   "Promo_code"      : "SNAP",   "Promo_desc"      : "Get 40% OFF",   "Promo_disc"      : 40.0,   "Promo_start"     : "2023-10-01",   "Promo_end"       : "2024-06-01" }</pre>
<pre>{   "Promo_code"      : "RAHMAH50",   "Promo_desc"      : "Get 50% OFF",   "Promo_disc"      : 50.0, </pre>	<pre>{   "Promo_code"      : "FLYSISWA",   "Promo_desc"      : "Get 70% OFF",   "Promo_disc"      : 70.0, </pre>

"Promo_start" : "2024-05-01", "Promo_end" : "2024-06-30" }	"Promo_start" : "2024-01-01", "Promo_end" : "2024-12-31" }
--	--

Table 4.4 10 sample documents for Promotion

Promotion 0.004 sec.						
	_id	Promo_code	Promo_desc	Promo_disc	Promo_start	Promo_end
1	ObjectId("6...	WELCOME10	Get 10% OFF	10.0	2024-04-01	2024-07-31
2	ObjectId("6...	WELCOME...	Get 20% OFF	20.0	2024-01-01	2024-09-01
3	ObjectId("6...	LOWFARE	Get 5% OFF	5.0	2024-03-01	2024-06-29
4	ObjectId("6...	MAYDEAL	Get 30% OFF	30.0	2024-03-01	2024-05-31
5	ObjectId("6...	RAHMAH50	Get 50% OFF	50.0	2024-05-01	2024-06-30
6	ObjectId("6...	BIGPAY5	Get 5% OFF	5.0	2024-06-01	2024-07-31
7	ObjectId("6...	HOLIDAY	Get 60% OFF	60.0	2024-03-01	2024-05-31
8	ObjectId("6...	WELCOME20	Get 20% OFF	20.0	2024-05-01	2024-07-30
9	ObjectId("6...	SNAP	Get 40% OFF	40.0	2023-10-01	2024-06-01
10	ObjectId("6...	FLYISWA	Get 70% OFF	70.0	2024-01-01	2024-12-31

Figure 4.4 Table view of the 10 sample documents for Promotion

- Payment

<pre>{   "Payment_id"    : 1234567890,   "Payment_method" : "Credit Card" } {   "Payment_id"    : 9876543210,   "Payment_method" : "Online Banking" } {   "Payment_id"    : 5678901234,   "Payment_method" : "Debit Card" } {   "Payment_id"    : 2345678901,   "Payment_method" : "Debit Card" } {   "Payment_id"    : 8765432109,   "Payment_method" : "BigPay" }</pre>	<pre>{   "Payment_id"    : 3456789012,   "Payment_method" : "BigPay" } {   "Payment_id"    : 9012345678,   "Payment_method" : "BigPay" } {   "Payment_id"    : 7890123456,   "Payment_method" : "Credit Card" } {   "Payment_id"    : 2109876543,   "Payment_method": "Online Banking" } {   "Payment_id"    : 6543210987,   "Payment_method" : "Debit Card" }</pre>
---	--

Table 4.5 10 sample documents for Payment

Payment 0.005 sec.			
	_id	Payment_id	Payment_method
1	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 1234567890	<input type="checkbox"/> Credit Card
2	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 9876543210	<input type="checkbox"/> Online Banking
3	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 5678901234	<input type="checkbox"/> Debit Card
4	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 2345678901	<input type="checkbox"/> Debit Card
5	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 8765432109	<input type="checkbox"/> BigPay
6	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 3456789012	<input type="checkbox"/> BigPay
7	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 9012345678	<input type="checkbox"/> BigPay
8	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 7890123456	<input type="checkbox"/> Credit Card
9	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 2109876543	<input type="checkbox"/> Online Banking
10	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> 6543210987	<input type="checkbox"/> Debit Card

Figure 4.5 Table view of the 10 sample documents for Payment

- Flight

<pre>{   "Flight_no"      : "AK123",   "Flight_origin"   : "KUL",   "Flight_destination": "DPS",   "Flight_departdate": "2024-07-15",   "Flight_departtime": "19:45:00",   "Flight_arrivaldate": "2024-07-15",   "Flight_arrivaltime": "22:45:00" }</pre>	<pre>{   "Flight_no"      : "EE556",   "Flight_origin"   : "KUL",   "Flight_destination": "ICN",   "Flight_departdate": "2024-12-22",   "Flight_departtime": "23:00:00",   "Flight_arrivaldate": "2024-12-23",   "Flight_arrivaltime": "06:30:00" }</pre>
<pre>{   "Flight_no"      : "QZ456",   "Flight_origin"   : "BNE",   "Flight_destination": "KUL",   "Flight_departdate": "2024-09-03",   "Flight_departtime": "07:00:00",   "Flight_arrivaldate": "2024-09-04",   "Flight_arrivaltime": "04:15:00" }</pre>	<pre>{   "Flight_no"      : "WW778",   "Flight_origin"   : "SYD",   "Flight_destination": "KUL",   "Flight_departdate": "2024-03-08",   "Flight_departtime": "11:15:00",   "Flight_arrivaldate": "2024-03-09",   "Flight_arrivaltime": "00:05:00" }</pre>
<pre>{   "Flight_no"      : "FD789",   "Flight_origin"   : "BWN",   "Flight_destination": "KUL",   "Flight_departdate": "2024-05-28",   "Flight_departtime": "16:10:00",   "Flight_arrivaldate": "2024-05-28",   "Flight_arrivaltime": "18:35:00" }</pre>	<pre>{   "Flight_no"      : "PQ890",   "Flight_origin"   : "KUL",   "Flight_destination": "KIX",   "Flight_departdate": "2024-08-12",   "Flight_departtime": "01:55:00",   "Flight_arrivaldate": "2024-08-12",   "Flight_arrivaltime": "09:35:00" }</pre>
<pre>{   "Flight_no"      : "XT234",   "Flight_origin"   : "KUL",   "Flight_destination": "MNL",   "Flight_departdate": "2024-10-10",</pre>	<pre>{   "Flight_no"      : "BB112",   "Flight_origin"   : "KUL",   "Flight_destination": "TPE",   "Flight_departdate": "2024-11-05",</pre>

<pre> "Flight_departtime" : "08:00:00", "Flight_arrivaldate": "2024-10-10", "Flight_arrivaltime": "12:15:00" } {   "Flight_no"      : "Z2567",   "Flight_origin"   : "ICN",   "Flight_destination": "KUL",   "Flight_departdate": "2024-06-20",   "Flight_departtime": "18:45:00",   "Flight_arrivaldate": "2024-06-21",   "Flight_arrivaltime": "00:15:00" } </pre>	<pre> "Flight_departtime" : "09:50:00", "Flight_arrivaldate": "2024-11-05", "Flight_arrivaltime": "14:40:00" } {   "Flight_no"      : "CC334",   "Flight_origin"   : "KUL",   "Flight_destination": "SIN",   "Flight_departdate": "2024-04-17",   "Flight_departtime": "06:05:00",   "Flight_arrivaldate": "2024-04-17",   "Flight_arrivaltime": "07:15:00" } </pre>
--	--

Table 4.6 10 sample documents for Flight

Flight 0.005 sec.								
	_id	Flight_no	Flight_origin	Flight_destination	Flight_departdate	Flight_departtime	Flight_arrivaldate	Flight_arrivaltime
1	ObjectId("6...	AK123	KUL	DPS	2024-07-15	19:45:00	2024-07-15	22:45:00
2	ObjectId("6...	QZ456	BNE	KUL	2024-09-03	07:00:00	2024-09-04	04:15:00
3	ObjectId("6...	FD789	BWN	KUL	2024-05-28	16:10:00	2024-05-28	18:35:00
4	ObjectId("6...	XT234	KUL	MNL	2024-10-10	08:00:00	2024-10-10	12:15:00
5	ObjectId("6...	Z2567	ICN	KUL	2024-06-20	18:45:00	2024-06-21	00:15:00
6	ObjectId("6...	PQ890	KUL	KIX	2024-08-12	01:55:00	2024-08-12	09:35:00
7	ObjectId("6...	BB112	KUL	TPE	2024-11-05	09:50:00	2024-11-05	14:40:00
8	ObjectId("6...	CC334	KUL	SIN	2024-04-17	06:05:00	2024-04-17	07:15:00
9	ObjectId("6...	EE556	KUL	ICN	2024-12-22	23:00:00	2024-12-23	06:30:00
10	ObjectId("6...	WW778	SYD	KUL	2024-03-08	11:15:00	2024-03-09	00:05:00

Figure 4.6 Table view of the 10 sample documents for Flight

- Time

<pre>{   "Time_id"      : 1679802000,   "Time_quater"  : 2,   "Time_month"   : "April",   "Time_day"     : "Saturday",   "Time_date"    : "2024-04-20" }</pre>	<pre>{   "Time_id"      : 1692656400,   "Time_quater"  : 1,   "Time_month"   : "March",   "Time_day"     : "Monday",   "Time_date"    : "2024-03-18" }</pre>
<pre>{   "Time_id"      : 1697758800,   "Time_quater"  : 2,   "Time_month"   : "May",   "Time_day"     : "Saturday",   "Time_date"    : "2024-05-11" }</pre>	<pre>{   "Time_id"      : 1690947600,   "Time_quater"  : 2,   "Time_month"   : "June",   "Time_day"     : "Tuesday",   "Time_date"    : "2024-06-25" }</pre>
<pre>{   "Time_id"      : 1684218000,   "Time_quater"  : 1,   "Time_month"   : "January",   "Time_day"     : "Tuesday",   "Time_date"    : "2024-01-02" }</pre>	<pre>{   "Time_id"      : 1694910800,   "Time_quater"  : 4,   "Time_month"   : "December",   "Time_day"     : "Friday",   "Time_date"    : "2023-12-23" }</pre>
<pre>{   "Time_id"      : 1698997200,   "Time_quater"  : 2,   "Time_month"   : "June",   "Time_day"     : "Sunday",   "Time_date"    : "2024-06-30" }</pre>	<pre>{   "Time_id"      : 1689541200,   "Time_quater"  : 3,   "Time_month"   : "July",   "Time_day"     : "Tuesday",   "Time_date"    : "2024-07-09" }</pre>
<pre>{   "Time_id"      : 1696890000,   "Time_quater"  : 1,   "Time_month"   : "March", </pre>	<pre>{   "Time_id"      : 1681664400,   "Time_quater"  : 4,   "Time_month"   : "November", </pre>

"Time_day" : "Thursday", "Time_date" : "2024-03-07" }	"Time_day" : "Wednesday", "Time_date" : "2023-11-29" }
---	--

Table 4.7 10 sample documents for Time

Time 0.004 sec.						
	_id	Time_id	Time_quater	Time_month	Time_day	Time_date
1	ObjectId("6...	1679802000	2	April	Saturday	2024-04-20
2	ObjectId("6...	1697758800	2	May	Saturday	2024-05-11
3	ObjectId("6...	1684218000	1	January	Tuesday	2024-01-02
4	ObjectId("6...	1698997200	2	June	Sunday	2024-06-30
5	ObjectId("6...	1692656400	1	March	Monday	2024-03-18
6	ObjectId("6...	1696890000	1	March	Thursday	2024-03-07
7	ObjectId("6...	1690947600	2	June	Tuesday	2024-06-25
8	ObjectId("6...	1694910800	4	December	Friday	2023-12-23
9	ObjectId("6...	1689541200	3	July	Tuesday	2024-07-09
10	ObjectId("6...	1681664400	4	November	Wednesday	2023-11-29

Figure 4.7 Table view of the 10 sample documents for Time

- Sales

{ "Sales_id" : "AB123456", "Cust_id" : 10077, "Loc_id" : "KUL", "Time_id" : 1679802000, "Company_id" : "AWQ", "Promo_code" : "WELCOME10", "Payment_id" : 1234567890, "Flight_no" : "AK123", "Sale_total" : 329.00 }	{ "Sales_id" : "QR901234", "Cust_id" : 90567, "Loc_id" : "KUL", "Time_id" : 1689541200, "Company_id" : "AXM", "Promo_code" : "WELCOME20", "Payment_id" : 2109876543, "Flight_no" : "EE556", "Sale_total" : 850.00 }
{ "Sales_id" : "CD789012", "Cust_id" : 20012, "Loc_id" : "BNE", }	{ "Sales_id" : "ST567890", "Cust_id" : 11384, "Loc_id" : "SYD", }



"Time_id" : 1697758800, "Company_id" : "XAX", "Promo_code" : "LOWFARE", "Payment_id" : 9876543210, "Flight_no" : "QZ456", "Sale_total" : 1524.00 } { "Sales_id" : "EF345678", "Cust_id" : 30456, "Loc_id" : "BWN", "Time_id" : 1684218000, "Company_id" : "AXM", "Promo_code" : "WELCOMEAPP", "Payment_id" : 5678901234, "Flight_no" : "FD789", "Sale_total" : 1435.00 } { "Sales_id" : "MN789012", "Cust_id" : 70897, "Loc_id" : "KUL", "Time_id" : 1690947600, "Company_id" : "TAX", "Promo_code" : "BIGPAY5", "Payment_id" : 9012345678, "Flight_no" : "BB112", "Sale_total" : 415.00 }} { "Sales_id" : "OP345678", "Cust_id" : 77239, "Loc_id" : "KUL", "Time_id" : 1694910800,	"Time_id" : 1681664400, "Company_id" : "XAX", "Promo_code" : "SNAP", "Payment_id" : 6543210987, "Flight_no" : "WW778", "Sale_total" : 1825.00 } { "Sales_id" : "GH901234", "Cust_id" : 40123, "Loc_id" : "KUL", "Time_id" : 1698997200, "Company_id" : "APG", "Promo_code" : "FLYSISWA", "Payment_id" : 2345678901, "Flight_no" : "XT234", "Sale_total" : 422.00 } { "Sales_id" : "IJ567890", "Cust_id" : 50234, "Loc_id" : "ICN", "Time_id" : 1692656400, "Company_id" : "KTC", "Promo_code" : "RAHMAH50", "Payment_id" : 8765432109, "Flight_no" : "Z2567", "Sale_total" : 522.00 }} { "Sales_id" : "KL123456", "Cust_id" : 60567, "Loc_id" : "KUL", "Time_id" : 1696890000,
--	---

"Company_id" : "AXM",	"Company_id" : "WAJ",
"Promo_code" : "SNAP",	"Promo_code" : "RAHMAH50",
"Payment_id" : 7890123456,	"Payment_id" : 3456789012,
"Flight_no" : "CC334",	"Flight_no" : "PQ890",
"Sale_total" : 200.00	"Sale_total" : 923.00
}	}

Table 4.8 10 sample documents for Sales

	_id	Sales_id	Cust_id	Loc_id	Time_id	Company_id	Promo_code	Payment_id	Flight_no	Sale_total
1	<div>Objectid("6...</div>	AB123456	<div>10077</div>	<div>KUL</div>	<div>1679802000</div>	<div>AWQ</div>	<div>WELCOME10</div>	<div>1234567890</div>	<div>AK123</div>	<div>329.0</div>
2	<div>Objectid("6...</div>	CD789012	<div>20012</div>	<div>BNE</div>	<div>1697758800</div>	<div>XAX</div>	<div>LOWFARE</div>	<div>9876543210</div>	<div>QZ456</div>	<div>1524.0</div>
3	<div>Objectid("6...</div>	EF345678	<div>30456</div>	<div>BWN</div>	<div>1684218000</div>	<div>AXM</div>	<div>WELCOME...</div>	<div>5678901234</div>	<div>FD789</div>	<div>1435.0</div>
4	<div>Objectid("6...</div>	GH901234	<div>40123</div>	<div>KUL</div>	<div>1698997200</div>	<div>APG</div>	<div>FLYSISWA</div>	<div>2345678901</div>	<div>XT234</div>	<div>422.0</div>
5	<div>Objectid("6...</div>	IJ567890	<div>50234</div>	<div>ICN</div>	<div>1692656400</div>	<div>KTC</div>	<div>RAHMAH50</div>	<div>8765432109</div>	<div>Z2567</div>	<div>522.0</div>
6	<div>Objectid("6...</div>	KL123456	<div>60567</div>	<div>KUL</div>	<div>1696890000</div>	<div>WAJ</div>	<div>RAHMAH50</div>	<div>3456789012</div>	<div>PQ890</div>	<div>923.0</div>
7	<div>Objectid("6...</div>	MN789012	<div>70897</div>	<div>KUL</div>	<div>1690947600</div>	<div>TAX</div>	<div>BIGPAY5</div>	<div>9012345678</div>	<div>BB112</div>	<div>415.0</div>
8	<div>Objectid("6...</div>	OP345678	<div>77239</div>	<div>KUL</div>	<div>1694910800</div>	<div>AXM</div>	<div>SNAP</div>	<div>7890123456</div>	<div>CC334</div>	<div>200.0</div>
9	<div>Objectid("6...</div>	QR901234	<div>90567</div>	<div>KUL</div>	<div>1689541200</div>	<div>AXM</div>	<div>WELCOME20</div>	<div>2109876543</div>	<div>EE556</div>	<div>850.0</div>
10	<div>Objectid("6...</div>	ST567890	<div>11384</div>	<div>SYD</div>	<div>1681664400</div>	<div>XAX</div>	<div>SNAP</div>	<div>6543210987</div>	<div>WW778</div>	<div>1825.0</div>

Figure 4.7 Table view of the 10 sample documents for Time

## 5.0 Complex Query

### 5.1 Logical Operation

This query will return sales documents where:

- The Sale\_total is less than 1000.
- The Cust\_gender is "MALE" or the Promo\_code starts with "WELCOME".
- The Company\_id is not "AIQ" or "TAX".

- NoSQL Command

```
db.Sales.find({
  $and: [
    {
      Sale_total: {
        $lt: 1000
      }
    },
    {
      $or: [
        {
          Cust_gender: {
            $exists: true,
            $eq: "MALE"
          }
        },
        {
          Promo_code: {
            $regex: /^WELCOME/
          }
        }
      ]
    },
    {
      Company_id: {
        $not: {
          $in: ["AIQ", "TAX"]
        }
      }
    }
  ]
})
```

Figure 5.0 Query for Logical Operation OR, AND, NOT

- Output

Sales 0.001 sec.										
	_id	Sales_id	Cust_id	Loc_id	Time_id	Company_id	Promo_code	Payment_id	Flight_no	Sale_total
1	ObjectId("6...)	AB123456	10077	KUL	1679802000	AWQ	WELCOME10	1234567890	AK123	329.0
2	ObjectId("6...)	QR901234	90567	KUL	1689541200	AXM	WELCOME20	2109876543	EE556	850.0

Figure 5.1 Output for Logical Operation OR, AND, NOT

## 5.2 Comparison Operator

This query will return sales documents where:

- The Cust\_age is greater than or equal to 30.
- Sale\_total greater than 500.
- Promo\_code starts with ["RAHMAH"].
- Flight depart date that is greater than or equal to “2024-01-01” but lower than “2025-01-01”

- NoSQL Command

```
db.Sales.find({
  $and: [
    { Sale_total: { $gt: 500 } },
    {
      $or: [
        {
          $and: [
            { Cust_age: { $exists: true, $gte: 30 } },
            { Cust_gender: "MALE" },
          ]
        },
        { Promo_code: { $regex: /^RAHMAH/ } }
      ]
    },
    { Flight_departdate: { $gte: new Date("2024-01-01"), $lt: new Date("2025-01-01") } }
  ]
})
```

Figure 5.2 Query for Comparison Operator (greater than, greater than or equal to)

- Output

Sales 0.001 sec.										0
	_id	Sales_id	Cust_id	Loc_id	Time_id	Company_id	Promo_code	Payment_id	Flight_no	Sale_total
1	<input type="checkbox"/> ObjectId("6...	U567890	<input type="checkbox"/> 50234	<input type="checkbox"/> ICN	<input type="checkbox"/> 1692656400	<input type="checkbox"/> KTC	<input type="checkbox"/> RAHMAH50	<input type="checkbox"/> 8765432109	<input type="checkbox"/> Z2567	<input type="text" value="522.0"/>
2	<input type="checkbox"/> ObjectId("6...	KL123456	<input type="checkbox"/> 60567	<input type="checkbox"/> KUL	<input type="checkbox"/> 1696890000	<input type="checkbox"/> WAJ	<input type="checkbox"/> RAHMAH50	<input type="checkbox"/> 3456789012	<input type="checkbox"/> PQ890	<input type="text" value="923.0"/>

Figure 5.3 Output for Comparison Operator (greater than, greater than or equal to)

## 5.3 Aggregate Function

This aggregate function fetches sales data by Sales and Company collection, providing insights into total, average, transaction count for each sales along with the highest and lowest sales figures recorded for each company.

- NoSQL Command

```
db.getCollection("Sales").aggregate([
  {
    $lookup: {
      from: "Company",
      localField: "Company_id",
      foreignField: "Company_id",
      as: "company"
    }
  },
  { $unwind: "$company" },
  {
    $group: {
      _id: "$company.Company_name",
      totalSales: { $sum: "$Sale_total" },
      averageSale: { $avg: "$Sale_total" },
      maxSale: { $max: "$Sale_total" },
      minSale: { $min: "$Sale_total" },
      countSales: { $sum: 1 }
    }
  },
  {
    $project: {
      _id: 0,
      Company_name: "$_id",
      totalSales: 1,
      averageSale: 1,
      maxSale: 1,
      minSale: 1,
      countSales: 1
    }
  }
])
```

- Output

Sales 0.014 sec.						
	totalSales	averageSale	maxSale	minSale	countSales	Company_name
1	0	329.0	329.0	329.0	1.0	AirAsia Indonesia
2	0	422.0	422.0	422.0	1.0	AirAsia Philippines
3	0	522.0	522.0	522.0	1.0	AirAsia Cambodia
4	0	828.3333333333333	1435.0	200.0	3.0	AirAsia
5	0	923.0	923.0	923.0	1.0	AirAsia Japan
6	0	1674.5	1825.0	1524.0	2.0	AirAsia X

## 5.4 Project Command

This Project Command combines data from Sales, Company and Location collection to produce a view that includes specific fields, making sure that 'Loc\_country' are linked correctly to each sale based on the 'Loc\_id'.

- NoSQL Command

```
db.getCollection("Sales").aggregate([
  {
    $lookup: {
      from: "Company",
      localField: "Company_id",
      foreignField: "Company_id",
      as: "Company"
    }
  },
  { $unwind: "$Company" },
  {
    $lookup: {
      from: "Location",
      localField: "Loc_id",
      foreignField: "Loc_id",
      as: "Location"
    }
  },
  { $unwind: "$Location" },
  {
    $project: {
      Sales_id: 1,
      Cust_id: 1,
      Company_id: 1,
      Flight_no: 1,
      Loc_id: 1,
      Loc_country: "$Location.Loc_country"
    }
  }
])
```

- Output

Sales 0.004 sec.							
	_id	Sales_id	Cust_id	Loc_id	Company_id	Flight_no	Loc_country
1	ObjectId("6...	AB123456	10077	KUL	AWQ	AK123	Malaysia
2	ObjectId("6...	CD789012	20012	BNE	XAX	QZ456	Australia
3	ObjectId("6...	EF345678	30456	BWN	AXM	FD789	Brunei
4	ObjectId("6...	GH901234	40123	KUL	APG	XT234	Malaysia
5	ObjectId("6...	IJ567890	50234	ICN	KTC	Z2567	South Korea
6	ObjectId("6...	KL123456	60567	KUL	WAJ	PQ890	Malaysia
7	ObjectId("6...	MN789012	70897	KUL	TAX	BB112	Malaysia
8	ObjectId("6...	OP345678	77239	KUL	AXM	CC334	Malaysia
9	ObjectId("6...	QR901234	90567	KUL	AXM	EE556	Malaysia
10	ObjectId("6...	ST567890	11384	SYD	XAX	WW778	Australia

## 5.5 Size Clause

This size clause query combines data from Sales, Company, Time and Flight to view company name, time id, flight origin and destination, sales total, where sales total more than RM1000.00.

- NoSQL Command

```
db.getCollection("Sales").aggregate([
  {
    $lookup: {
      from: "Company",
      localField: "Company_id",
      foreignField: "Company_id",
      as: "Company"
    }
  },
  { $unwind: "$Company" },
  {
    $group: {
      _id: {
        Sales_id: "$Sales_id",
        Company_id: "$Company_id"
      },
      Sales: { $first: "$$ROOT" }
    }
  },
  {
    $lookup: {
      from: "Time",
      localField: "Sales.Time_id",
      foreignField: "Time_id",
      as: "Time"
    }
  },
  { $unwind: "$Time" },
```

```
{
  $group: {
    _id: {
      Sales_id: "$sales.Sales_id",
      Company_id: "$sales.Company_id",
      Time_id: "$Time.Time_id"
    },
    Sales: { $first: "$$ROOT" }
  }
},
{
  $lookup: {
    from: "Flight",
    localField: "Sales.Sales.Flight_no",
    foreignField: "Flight_no",
    as: "Flight"
  }
},
```

```

{ $unwind: "$Flight" },

{
  $match: {
    "Sales.Sales.Sale_total": { $gt: 1000.0 }
  }
},

{
  $project: {
    _id: 0,
    Company_name: "$Sales.Sales.Company.Company_name",
    Time_id: "$Sales.Time.Time_id",
    Flight_origin: "$Flight.Flight_origin",
    Flight_destination: "$Flight.Flight_destination",
    Sale_total: "$Sales.Sales.Sale_total"
  }
}
})

```

- Output

Sales 0.01 sec.					
	Company_name	Time_id	Flight_origin	Flight_destination	Sale_total
1	AirAsia X	1697758800	BNE	KUL	1524.0
2	AirAsia X	1681664400	SYD	KUL	1825.0
3	AirAsia	1684218000	BWN	KUL	1435.0



## 6.0 NoSQL Command to Update Particular Record(s)

- Find documents meeting all three criteria: female, under 30, with a gmail address
- Update their contact number to a new standardized number
- Increment their age by 1
- NoSQL Command

```
db.Customer.updateMany(  
  {  
    Cust_gender: "FEMALE",  
    Cust_age: { $lt: 30 },  
    Cust_email: { $regex: "@gmail.com$" }  
  },  
  {  
    $set: { Cust_contactnum: "+60100000000" },  
    $inc: { Cust_age: 1 }  
  }  
)
```

Figure 6.0.1 Query for Update based on certain criteria

- Output:

Customer 0.007 sec.								
	_id	Cust_id	Cust_name	Cust_age	Cust_passport	Cust_gender	Cust_contactnum	Cust_email
1	ObjectId("6...)	10077	Emily Smith	25	AB123456	FEMALE	+15551234567	emilysm@gmail.com
2	ObjectId("6...)	20012	Aminah Azman	30	XY987654	FEMALE	+60134568923	aminahaaa@gmail.com
3	ObjectId("6...)	30456	Aiman Arif	42	PQ456789	MALE	+60142345678	aimanarf@gmail.com
4	ObjectId("6...)	40123	Jennifer Lee	19	CD789012	FEMALE	+60137777456	jenniferlee@yahoo.com
5	ObjectId("6...)	50234	Daniel Iman	36	EF234567	MALE	+60115968275	daniell@gmail.com
6	ObjectId("6...)	60567	Kumar Singh	50	GH345678	MALE	+60122323979	kumarsk@yahoo.com
7	ObjectId("6...)	70897	Lee Soon Ye	28	LJ391728	MALE	+442012322678	soonye@gmail.com
8	ObjectId("6...)	77239	Ahmad Ali	44	ND827631	MALE	+60145955657	ahmadali@gmail.com
9	ObjectId("6...)	90567	Sophia Adam	22	ST398734	FEMALE	+6121345678	sophiadm@gmail.com
10	ObjectId("6...)	11384	Wan Seri	53	AP493789	FEMALE	+60138955789	wanseri@gmail.com

Figure 6.0.2 Table view of the 10 sample documents for Customer before Update

	acknowledged	matchedCount	modifiedCount
1	T/F true	2.0	2.0

Figure 6.0.3 2 Affected fields

Customer 0.002 sec.								
	_id	Cust_id	Cust_name	Cust_age	Cust_passport	Cust_gender	Cust_contactnum	Cust_email
1	ObjectId("6...	10077	Emily Smith	26.0	AB123456	FEMALE	+6010000000	emilysm@gmail.com
2	ObjectId("6...	20012	Aminah Azman	30	XY987654	FEMALE	+60134568923	aminahaaa@gmail.com
3	ObjectId("6...	30456	Aiman Arif	42	PQ456789	MALE	+60142345678	aimanarif@gmail.com
4	ObjectId("6...	40123	Jennifer Lee	19	CD789012	FEMALE	+60137777456	jenniferlee@yahoo.com
5	ObjectId("6...	50234	Daniel Iman	36	EF234567	MALE	+60115968275	daniell@gmail.com
6	ObjectId("6...	60567	Kumar Singh	50	GH345678	MALE	+60122323979	kumarsk@yahoo.com
7	ObjectId("6...	70897	Lee Soon Ye	28	LJ391728	MALE	+442012322678	soonye@gmail.com
8	ObjectId("6...	77239	Ahmad Ali	44	ND827631	MALE	+60145955657	ahmadali@gmail.com
9	ObjectId("6...	90567	Sophia Adam	23.0	ST398734	FEMALE	+60100000000	sophiadm@gmail.com

Figure 6.0.4 Table view of the sample for Customer after Update

## 7.0 NoSQL Command to Delete Specific Record(s)

Delete documents where:

- The flight origin is Kuala Lumpur (KUL)
- The departure date is before June 1, 2024
- NoSQL Command

```
db.Flight.deleteMany(
{
  Flight_origin: "KUL",
  Flight_departdate: { $lt: "2024-11-01" }
})
```

Figure 7.0.1 Query for Delete based on certain criteria

Output:

Flight 0.005 sec.								
	_id	Flight_no	Flight_origin	Flight_destination	Flight_departdate	Flight_departtime	Flight_arrivaldate	Flight_arrivaltime
1	ObjectId("6...	AK123	KUL	DPS	2024-07-15	19:45:00	2024-07-15	22:45:00
2	ObjectId("6...	QZ456	BNE	KUL	2024-09-03	07:00:00	2024-09-04	04:15:00
3	ObjectId("6...	FD789	BWN	KUL	2024-05-28	16:10:00	2024-05-28	18:35:00
4	ObjectId("6...	XT234	KUL	MNL	2024-10-10	08:00:00	2024-10-10	12:15:00
5	ObjectId("6...	Z2567	ICN	KUL	2024-06-20	18:45:00	2024-06-21	00:15:00
6	ObjectId("6...	PQ890	KUL	KIX	2024-08-12	01:55:00	2024-08-12	09:35:00
7	ObjectId("6...	BB112	KUL	TPE	2024-11-05	09:50:00	2024-11-05	14:40:00
8	ObjectId("6...	CC334	KUL	SIN	2024-04-17	06:05:00	2024-04-17	07:15:00
9	ObjectId("6...	EE556	KUL	ICN	2024-12-22	23:00:00	2024-12-23	06:30:00
10	ObjectId("6...	WW778	SYD	KUL	2024-03-08	11:15:00	2024-03-09	00:05:00

Figure 7.0.2 Table view for Flight before Delete

	acknowledged	deletedCount
1	<input type="checkbox"/> T/F true	<input type="checkbox"/> ## 4.0

Figure 7.0.3 4 Affected Fields

Flight 0.001 sec.								
	_id	Flight_no	Flight_origin	Flight_destination	Flight_departdate	Flight_departtime	Flight_arrivaldate	Flight_arrivaltime
1	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> QZ456	<input type="checkbox"/> BNE	<input type="checkbox"/> KUL	<input type="checkbox"/> 2024-09-03	<input type="checkbox"/> 07:00:00	<input type="checkbox"/> 2024-09-04	<input type="checkbox"/> 04:15:00
2	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> FD789	<input type="checkbox"/> BWN	<input type="checkbox"/> KUL	<input type="checkbox"/> 2024-05-28	<input type="checkbox"/> 16:10:00	<input type="checkbox"/> 2024-05-28	<input type="checkbox"/> 18:35:00
3	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> Z2567	<input type="checkbox"/> ICN	<input type="checkbox"/> KUL	<input type="checkbox"/> 2024-06-20	<input type="checkbox"/> 18:45:00	<input type="checkbox"/> 2024-06-21	<input type="checkbox"/> 00:15:00
4	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> BB112	<input type="checkbox"/> KUL	<input type="checkbox"/> TPE	<input type="checkbox"/> 2024-11-05	<input type="checkbox"/> 09:50:00	<input type="checkbox"/> 2024-11-05	<input type="checkbox"/> 14:40:00
5	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> EE556	<input type="checkbox"/> KUL	<input type="checkbox"/> ICN	<input type="checkbox"/> 2024-12-22	<input type="checkbox"/> 23:00:00	<input type="checkbox"/> 2024-12-23	<input type="checkbox"/> 06:30:00
6	<input type="checkbox"/> ObjectId("6...	<input type="checkbox"/> WW778	<input type="checkbox"/> SYD	<input type="checkbox"/> KUL	<input type="checkbox"/> 2024-03-08	<input type="checkbox"/> 11:15:00	<input type="checkbox"/> 2024-03-09	<input type="checkbox"/> 00:05:00

Figure 7.0.2 Table view for Flight after Delete

## 8.0 Two NoSQL Command Not Covered in Lecture

### 8.1 Function findOneAndReplace()

- This function will first run a find function to retrieve the data from Customer with cust\_name of Alex Grace or cust\_age = 30 and cust\_gender = Male
- If data exists, replace it with new customer details
- If no data are found from the collection, replace it by creating a new customer document
- NoSQL Command

```
Local localhost:27017 AirAsia
db.Customer.findOneAndReplace(
  { $or: [
    { Cust_name: "Alex Grace" },
    { Cust_age: 30 }
  ] },
  {
    Cust_id: 87643,
    Cust_name: "Alex Grace",
    Cust_age: 35,
    Cust_passport: "DG879012",
    Cust_gender: "Male",
    Cust_contactnum: "+9876543210",
    Cust_email: "alexgrace@gmail.com",
    upsert: true,
    returnDocument: "after"
  }
);
```

- Output

Customer 0.004 sec.							
_id	Cust_id	Cust_name	Cust_age	Cust_passport	Cust_gender	Cust_contactnum	Cust_email
1	ObjectId("66... 10077	Emily Smith	25	AB123456	FEMALE	+155512345...	emilysm@g...
2	ObjectId("66... 20012	Aminah ...	30	XY987654	FEMALE	+601345689...	aminahaaa...
3	ObjectId("66... 30456	Aiman Arif	42	PQ456789	MALE	+601423456...	aimanarf@g...
4	ObjectId("66... 40123	Jennifer Lee	19	CD789012	FEMALE	+601377774...	jenniferlee@...
5	ObjectId("66... 50234	Daniel Iman	36	EF234567	MALE	+601159682...	daniell@gm...
6	ObjectId("66... 60567	Kumar Singh	50	GH345678	MALE	+601223239...	kumarsk@ya...
7	ObjectId("66... 70897	Lee Soon Ye	28	LJ391728	MALE	+442012322...	soonye@gm...
8	ObjectId("66... 77239	Ahmad Ali	44	ND827631	MALE	+601459556...	ahmadali@...
9	ObjectId("66... 90567	Sophia Adam	22	ST398734	FEMALE	+6121345678	sophiadm@...
10	ObjectId("66... 11384	Wan Seri	53	AP493789	FEMALE	+601389557...	wanseri@g...

Before execute the command

Customer 0.002 sec.							
_id	Cust_id	Cust_name	Cust_age	Cust_passport	Cust_gender	Cust_contactnum	Cust_email
1	ObjectId("66... 10077	Emily Smith	25	AB123456	FEMALE	+155512345...	emilysm@g...
2	ObjectId("66... 20012	Aminah ...	30	XY987654	FEMALE	+601345689...	aminahaaa...
3	ObjectId("66... 30456	Aiman Arif	42	PQ456789	MALE	+601423456...	aimanarf@g...
4	ObjectId("66... 40123	Jennifer Lee	19	CD789012	FEMALE	+601377774...	jenniferlee@...
5	ObjectId("66... 50234	Daniel Iman	36	EF234567	MALE	+601159682...	daniell@gm...
6	ObjectId("66... 60567	Kumar Singh	50	GH345678	MALE	+601223239...	kumarsk@ya...
7	ObjectId("66... 70897	Lee Soon Ye	28	LJ391728	MALE	+442012322...	soonye@gm...
8	ObjectId("66... 77239	Ahmad Ali	44	ND827631	MALE	+601459556...	ahmadali@...
9	ObjectId("66... 90567	Sophia Adam	22	ST398734	FEMALE	+6121345678	sophiadm@...
10	ObjectId("66... 11384	Wan Seri	53	AP493789	FEMALE	+601389557...	wanseri@g...
11	ObjectId("66... 87643.0	Alex Grace	35.0	DG879012	Male	+9876543210	alexgrace@...

After execute the command, customer named Alex Grace is added to the document

## 8.2 Function distinct()

- To retrieve distinct promo code active in 2024.
- NoSQL command

```
db.PROMOTION.distinct("Promo_code", {
  $or: [
    {
      "Promo_start": { $gte: "2024-01-01", $lte: "2024-12-31" }
    },
    {
      "Promo_end": { $gte: "2024-01-01", $lte: "2024-12-31" }
    },
  ]
})
```

Promotion 0.005 sec.						
	_id	Promo_code	Promo_desc	Promo_disc	Promo_start	Promo_end
1	ObjectId("66...	WELCOME10	Get 10% OFF	10.0	2024-04-01	2024-07-31
2	ObjectId("66...	WELCOMEAPP	Get 20% OFF	20.0	2024-01-01	2024-09-01
3	ObjectId("66...	LOWFARE	Get 5% OFF	5.0	2024-03-01	2024-06-29
4	ObjectId("66...	MAYDEAL	Get 30% OFF	30.0	2024-03-01	2024-05-31
5	ObjectId("66...	RAHMAH50	Get 50% OFF	50.0	2024-05-01	2024-06-30
6	ObjectId("66...	BIGPAY5	Get 5% OFF	5.0	2024-06-01	2024-07-31
7	ObjectId("66...	HOLIDAY	Get 60% OFF	60.0	2024-03-01	2024-05-31
8	ObjectId("66...	WELCOME20	Get 20% OFF	20.0	2024-05-01	2024-07-30
9	ObjectId("66...	SNAP	Get 40% OFF	40.0	2023-10-01	2024-06-01
10	ObjectId("66...	FLYSISWA	Get 70% OFF	70.0	2024-01-01	2024-12-31

Table view of the 10 sample documents for Promotion

- Output

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	BIGPAY5	FLYSISWA	HOLIDAY	LOWFARE	MAYDEAL	RAHMAH50	SNAP	WELCOME10	WELCOME20	WELCOMEAPP

## 9.0 Part B: Essay on The Impact of Big Data in Airlines Industry

### The impact of Big Data in Airlines Industry

Big data technology has been widely used in various industries nowadays, including Airlines industry. This technology requires advanced databases implementation like NoSQL to support the input of big data and improves performances. As airlines collecting a huge amount of data from various sources such as customers and operational side, this has impacts on how airlines operate, how marketing revenue are planned, enhances customer experiences, and improves safety of the flights.

Primarily, big data technology impacts the flight operational time by suggesting more efficient routes and schedules. For example, airlines can use data analytics to optimize flight paths, reducing travel time and fuel consumption. This enables them to create more accurate and efficient flight schedules, minimizing delays and improving punctuality. With this, airlines can manage fuel costs more effectively, enhancing both operational efficiency and cost savings.

Secondly, the use of big data has also impacted the marketing revenue planning of airlines. By analyzing sales trends, customer behaviours, and booking patterns, airlines can develop strategic plans to boost market sales. For example, they can create targeted offers and vouchers, personalize services, and implement dynamic pricing strategies. This approach helps airlines attract more customers, increase loyalty, and ultimately enhance their revenue.

Big data greatly improves customer experience by analyzing extensive customer data such as travel history and feedback. Airlines can provide personalized services, including customized travel suggestions and specific promotions. Analyzing data in real-time allows airlines to provide passengers immediate updates on flight status, gate changes, and travel plans, ensuring smoother and informed journey. By understanding customer preferences, airlines can enhance onboard services, thereby increasing passenger satisfaction.

Lastly, big data technologies that track and analyze data from aircraft sensors and maintenance logs through predictive analytics will have a significant impact on improving flight safety and maintenance. By identifying anomalies, airlines can predict potential equipment failures and plan preventive maintenance, reducing the risk of in-flight issues. Data-driven insights optimize maintenance schedules, reduce aircraft downtime, and improve fleet reliability. This proactive strategy ensures the passengers and crew safety while maintaining operational efficiency.

Conclusively, big data significantly enhances the airline industry by optimizing operations, refining marketing revenue strategies, improving customer experiences, and ensuring flight safety. This data-driven approach drives efficiency, cost savings, and passenger satisfaction, revolutionizing how airlines operate and compete. As technology continues to evolve, the potential for further advancements in data analytics will likely lead to even greater innovations and improvements in the industry.

[Total words: 400]

Citation/Ref:

- [1] *Big data in airport Optimizaiton* / TAV Technologies. (2024, May 10). TAV Technologies. <https://tavtechnologies.aero/en-EN/review/pages/big-data-role-in-ariport-optimization>
- [2] Terekhov, V. (2024, May 20). *Revolutionizing Flight: Big data aviation software development*. Attract Group. <https://attractgroup.com/blog/big-data-aviation-software-development/#:~:text=Big%20data%20plays%20a%20critical,critical%20for%20maintaining%20flight%20safety>.
- [3] *Big data in the airline industry* / LinkedIn. (2022, April 22). <https://www.linkedin.com/pulse/big-data-airline-industry-busam-divya-prakash/>
- [4] *What are the most innovative ways to optimize airline operations using data and analytics?* (n.d.). <https://www.linkedin.com/advice/0/what-most-innovative-ways-optimize-airline-operations>
- [5] *Big Data Analytics and an Intelligent Aviation Information Management System*. [https://www.researchgate.net/publication/351884057\\_Big\\_Data\\_Analytics\\_and\\_an\\_Intelligent\\_Aviation\\_Information\\_Management\\_System](https://www.researchgate.net/publication/351884057_Big_Data_Analytics_and_an_Intelligent_Aviation_Information_Management_System)
- [6] *Big data analytics in airlines: Opportunities and Challenges*. [https://www.researchgate.net/publication/356647231\\_Big\\_data\\_analytics\\_in\\_airlines\\_Opportunities\\_and\\_Challenges](https://www.researchgate.net/publication/356647231_Big_data_analytics_in_airlines_Opportunities_and_Challenges)