



ULUSLARARASI KIBRIS ÜNİVERSİTESİ  
CYPRUS INTERNATIONAL UNIVERSITY

**CMPE343 Project:**  
**Flight DATABASE**  
**MANAGEMENT**  
**SYSTEM**

By

AKALLAL KARIM

22321368

ISMAIL LAHLOU

22321060

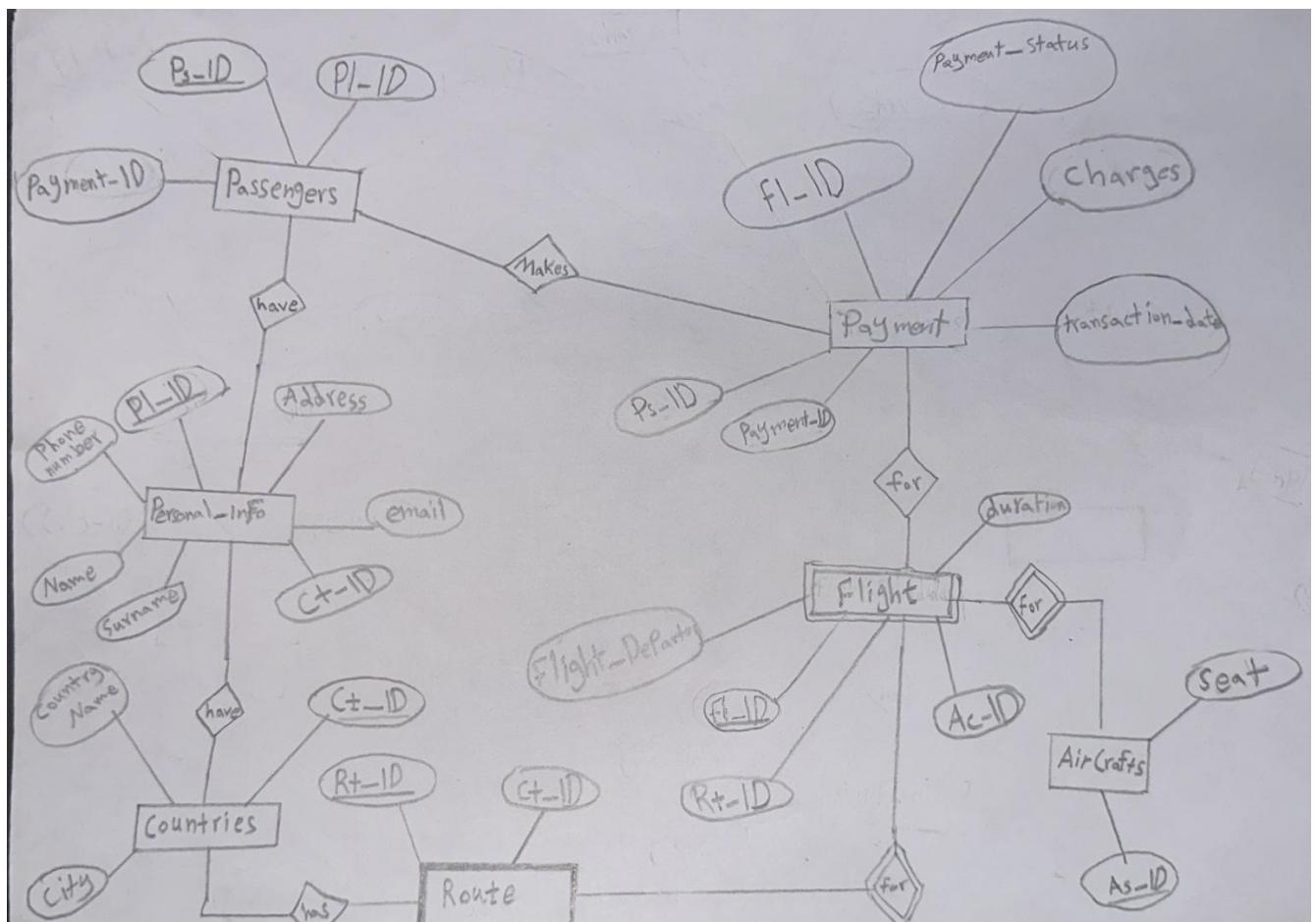
SALEM ALAWBATHANI

22211842

Marvellous Ogheneyole Alex

22512910

## 1. ER DIAGRAM



## 2. Creating Tables

```
CREATE TABLE Countries (
    ct_ID INT PRIMARY KEY,
    country_name VARCHAR(50) NOT NULL,
    city VARCHAR(50)
);
```

```
CREATE TABLE Route (
    rt_ID INT PRIMARY KEY,
    ct_ID INT,
    FOREIGN KEY (ct_ID) REFERENCES Countries(ct_ID)
);
```

```
CREATE TABLE AirCrafts (
    ac_ID INT PRIMARY KEY,
    seat INT
);
```

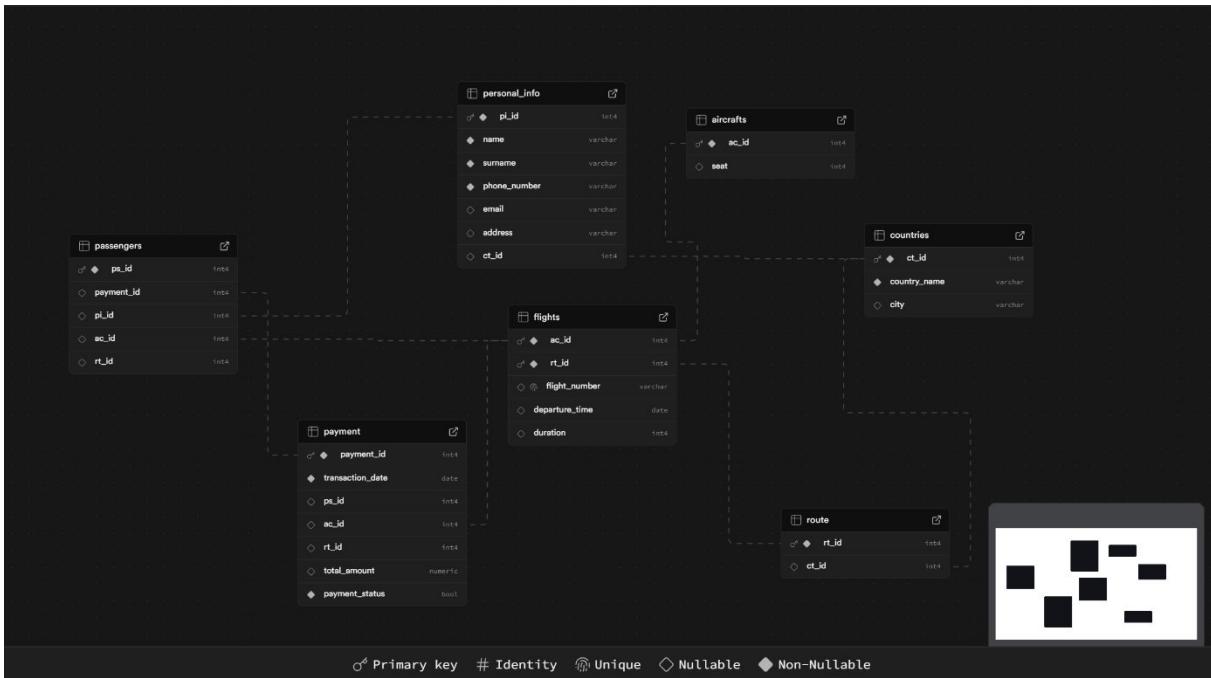
```
CREATE TABLE Flights (
    ac_ID INT,
    rt_ID INT,
    flight_number VARCHAR(10) unique,
    departure_time DATE,
    duration INT,
    CONSTRAINT fl_ID PRIMARY KEY (ac_ID, rt_ID),
    FOREIGN KEY (ac_ID) REFERENCES AirCrafts(ac_ID),
    FOREIGN KEY (rt_ID) REFERENCES Route(rt_ID));

```

```
CREATE TABLE Personal_info (
    pi_ID INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    surname VARCHAR(100) NOT NULL,
    phone_number VARCHAR(20) NOT NULL,
    email VARCHAR(100),
    address VARCHAR(255),
    ct_ID INT,
    FOREIGN KEY (ct_ID) REFERENCES Countries(ct_ID)
);
```

```
CREATE TABLE Passengers (
    ps_ID INT PRIMARY KEY,
    payment_id int,
    pi_ID INT,
    ac_ID INT,
    rt_ID INT,
    FOREIGN KEY (payment_id) REFERENCES payment(payment_id)
    FOREIGN KEY (pi_ID) REFERENCES Personal_info(pi_ID),
    FOREIGN KEY (ac_ID, rt_ID) REFERENCES Flights(ac_ID, rt_ID)
);

CREATE TABLE payment (
    payment_ID INT PRIMARY KEY,
    transaction_date DATE NOT NULL,
    ps_ID INT,
    ac_ID INT,
    rt_ID INT,
    total_amount numeric(10, 2),
    payment_status boolean default false,
    FOREIGN KEY (ps_ID) REFERENCES Passengers(ps_ID),
    FOREIGN KEY (ac_ID, rt_ID) REFERENCES Flights(ac_ID, rt_ID )
);
```



### 3. Filling tables:

- Countries table:

```

1  INSERT INTO Countries (ct_ID, country_name, city) VALUES
2  (1, 'United States', 'New York'),
3  (2, 'United Kingdom', 'London'),
4  (3, 'France', 'Paris'),
5  (4, 'Germany', 'Berlin'),
6  (5, 'Japan', 'Tokyo'),
7  (6, 'Australia', 'Sydney'),
8  (7, 'Canada', 'Toronto'),
9  (8, 'Italy', 'Rome'),
10 (9, 'Spain', 'Madrid'),
11 (10, 'Brazil', 'Rio de Janeiro');
12
13 select * from countries;

```

<b>ct_id</b>	<b>country_name</b>	<b>city</b>
1	United States	New York
2	United Kingdom	London
3	France	Paris
4	Germany	Berlin
5	Japan	Tokyo
6	Australia	Sydney
7	Canada	Toronto
8	Italy	Rome
9	Spain	Madrid
10	Brazil	Rio de Janeiro

### - **Personal\_info table:**

```

1  INSERT INTO Personal_info (pi_ID, name, surname, phone_number, email, address, ct_ID) VALUES
2  (1001, 'John', 'Smith', '+1-555-0101', 'john.smith@email.com', '123 Main St, NY', 1),
3  (1002, 'Emma', 'Johnson', '+44-20-7946-0958', NULL, '45 Baker St, London', 2),
4  (1003, 'Pierre', 'Dubois', '+33-1-2345-6789', 'pierre.d@email.com', NULL, 3),
5  (1004, 'Hans', 'Schmidt', '+49-30-123456', NULL, NULL, 4),
6  (1005, 'Yuki', 'Tanaka', '+81-3-1234-5678', 'yuki.t@email.com', 'Shibuya 2-1, Tokyo', 5),
7  (1006, 'Sarah', 'Wilson', '+61-2-9876-5432', NULL, 'George St 100, Sydney', 6),
8  (1007, 'David', 'Brown', '+1-416-555-1234', 'david.b@email.com', NULL, 7),
9  (1008, 'Giovanni', 'Rossi', '+39-06-1234567', NULL, 'Via del Corso 15, Rome', 8),
10 (1009, 'Maria', 'Garcia', '+34-91-2345678', 'maria.g@email.com', NULL, 9),
11 (1010, 'Carlos', 'Silva', '+55-21-98765-4321', NULL, NULL, 10);
12
13 select * from personal_info;

```

pi_id	name	surname	phone_number	email	address	ct_id
1001	John	Smith	+1-555-0101	john.smith@email.com	123 Main St, NY	1
1002	Emma	Johnson	+44-20-7946-0958	NULL	45 Baker St, London	2
1003	Pierre	Dubois	+33-1-2345-6789	pierre.d@email.com	NULL	3
1004	Hans	Schmidt	+49-30-123456	NULL	NULL	4
1005	Yuki	Tanaka	+81-3-1234-5678	yuki.t@email.com	Shibuya 2-1, Tokyo	5
1006	Sarah	Wilson	+61-2-9876-5432	NULL	George St 100, Sydney	6
1007	David	Brown	+1-416-555-1234	david.b@email.com	NULL	7
1008	Giovanni	Rossi	+39-06-1234567	NULL	Via del Corso 15, Rome	8
1009	Maria	Garcia	+34-91-2345678	maria.g@email.com	NULL	9
1010	Carlos	Silva	+55-21-98765-4321	NULL	NULL	10

- **Route table:**

```

1  INSERT INTO Route (rt_ID, ct_ID) VALUES
2  (101, 1), -- New York route
3  (102, 2), -- London route
4  (103, 3), -- Paris route
5  (104, 4), -- Berlin route
6  (105, 5), -- Tokyo route
7  (106, 6), -- Sydney route
8  (107, 7), -- Toronto route
9  (108, 8), -- Rome route
10 (109, 9), -- Madrid route
11 (110, 10); -- Rio route
12
13 select * from route

```

rt_id	ct_id
101	1
102	2
103	3
104	4
105	5
106	6
107	7
108	8
109	9
110	10

- Aircraft table:

```
1  INSERT INTO AirCrafts (ac_ID, seat) VALUES
2  (501, 180),
3  (502, 250),
4  (503, 300),
5  (504, 420),
6  (505, 150),
7  (506, 280),
8  (507, 320),
9  (508, 200),
10 (509, 350),
11
12 select * from Aircrafts;
```

rt_id	ct_id
101	1
102	2
103	3
104	4
105	5
106	6
107	7
108	8
109	9
110	10

## - Flight table:

```
1  INSERT INTO Flights (ac_ID, rt_ID, flight_number, departure_time, duration) VALUES
2  (501, 101, 'AA101', '2024-03-15 08:00:00', 420),
3  (502, 102, 'BA202', '2024-03-15 10:30:00', 480),
4  (503, 103, 'AF303', '2024-03-15 12:45:00', 90),
5  (504, 104, 'LH404', '2024-03-15 14:15:00', 120),
6  (505, 105, 'JL505', '2024-03-15 16:30:00', 780),
7  (506, 106, 'QF606', '2024-03-15 18:45:00', 960),
8  (507, 107, 'AC707', '2024-03-15 20:00:00', 360),
9  (508, 108, 'AZ808', '2024-03-15 22:15:00', 150),
10 (509, 109, 'IB909', '2024-03-16 06:00:00', 135);
11
12 select * from flights;
```

ac_id	rt_id	flight_number	departure_time	duration
501	101	AA101	2024-03-15	420
502	102	BA202	2024-03-15	480
503	103	AF303	2024-03-15	90
504	104	LH404	2024-03-15	120
505	105	JL505	2024-03-15	780
506	106	QF606	2024-03-15	960
507	107	AC707	2024-03-15	360
508	108	AZ808	2024-03-15	150
509	109	IB909	2024-03-16	135

- Payment table:

```
1  INSERT INTO payment (payment_ID, transaction_date, ps_ID, ac_ID, rt_ID, total_amount, payment_status) VALUES
2  (2001, '2024-03-01', 3001, 501, 101, 450.00, true),
3  (2002, '2024-03-02', 3002, 502, 102, 380.50, true),
4  (2003, '2024-03-03', 3003, 503, 103, 125.00, false),
5  (2004, '2024-03-04', 3004, 504, 104, 290.75, true),
6  (2005, '2024-03-05', 3005, 505, 105, 1200.00, true),
7  (2006, '2024-03-06', 3006, 506, 106, 950.25, true),
8  (2007, '2024-03-07', 3007, 507, 107, 320.00, false),
9  (2008, '2024-03-08', 3008, 508, 108, 180.00, true),
10 (2009, '2024-03-09', 3009, 509, 109, 210.50, true);
11
12
13 select * from payment;
```

payment_id	transaction_c	ps_id	ac_id	rt_id	total_amount	payment_status
2001	2024-03-01	3001	501	101	450.00	true
2002	2024-03-02	3002	502	102	380.50	true
2003	2024-03-03	3003	503	103	125.00	false
2004	2024-03-04	3004	504	104	290.75	true
2005	2024-03-05	3005	505	105	1200.00	true
2006	2024-03-06	3006	506	106	950.25	true
2007	2024-03-07	3007	507	107	320.00	false
2008	2024-03-08	3008	508	108	180.00	true
2009	2024-03-09	3009	509	109	210.50	true

- Passengers table:

```
1  INSERT INTO Passengers (ps_ID, payment_id, pi_ID, ac_ID, rt_ID) VALUES
2  (3001, 2001, 1001, 501, 101),
3  (3002, 2002, 1002, 502, 102),
4  (3003, 2003, 1003, 503, 103),
5  (3004, 2004, 1004, 504, 104),
6  (3005, 2005, 1005, 505, 105),
7  (3006, 2006, 1006, 506, 106),
8  (3007, 2007, 1007, 507, 107),
9  (3008, 2008, 1008, 508, 108),
10 (3009, 2009, 1009, 509, 109);
11
12
13 select * from passengers;
```

ps_id	payment_id	pi_id	ac_id	rt_id
3001	2001	1001	501	101
3002	2002	1002	502	102
3003	2003	1003	503	103
3004	2004	1004	504	104
3005	2005	1005	505	105
3006	2006	1006	506	106
3007	2007	1007	507	107
3008	2008	1008	508	108
3009	2009	1009	509	109

## 4. Data manipulation queries:

- Updating table:

Passenger changing destination and aircraft

```
1 UPDATE passengers
2 set rt_id=105 , ac_id=505
3
4 where ps_id=3003;
5
6 select * from passengers
7 where ps_id=3003;
8
```

ps_id	payment_id	pi_id	ac_id	rt_id
3003	2003	1003	505	105

- Deleting row:

Passenger cancelling a flight booking

```
1 delete from passengers
2
3 where ps_id=3004;
4
5
6 select * from passengers;
7
```

ps_id	payment_id	pi_id	ac_id	rt_id
3001	2001	1001	501	101
3002	2002	1002	502	102
3005	2005	1005	505	105
3006	2006	1006	506	106
3007	2007	1007	507	107
3008	2008	1008	508	108
3009	2009	1009	509	109
3003	2003	1003	505	105

- Queries to sort personal info in ascending order

```

1  SELECT name,surname
2  from personal_info
3  ORDER BY surname, name ;

```

name	surname
David	Brown
Pierre	Dubois
Maria	Garcia
Emma	Johnson
Giovanni	Rossi
Hans	Schmidt
Carlos	Silva
John	Smith
Yuki	Tanaka
Sarah	Wilson

- Queries to select a specific passenger and his personal information

```

SELECT ps_id, pi.name,pi.surname,f.flight_number, c.country_name
FROM Personal_info pi
INNER JOIN Passengers p ON pi.pi_ID = p.pi_ID
INNER JOIN Flights f ON p.ac_ID = f.ac_ID AND p.rt_ID = f.rt_ID
INNER JOIN Route r ON f.rt_ID = r.rt_ID
INNER JOIN Countries c ON r.ct_ID = c.ct_ID
where ps_id=3003;

```

ps_id	name	surname	flight_number	country_name
3003	Pierre	Dubois	JL505	Japan

- Queries to select countries with more than 2 passengers.

```

1  SELECT c.country_name, COUNT(p.ps_ID) as passenger_count
2  FROM Countries c
3  INNER JOIN Personal_info pi ON c.ct_ID = pi.ct_ID
4  INNER JOIN Passengers p ON pi.pi_ID = p.pi_ID
5  GROUP BY c.country_name
6  HAVING COUNT(p.ps_ID) > 2
7  ORDER BY passenger_count DESC;

```

- Queries to select passenger's personal info from countries name have 'United'

```

1  SELECT pi.name, pi.surname, c.country_name
2  FROM Personal_info pi
3  INNER JOIN Countries c ON pi.ct_ID = c.ct_ID
4  WHERE c.country_name like '%United%';

```

<b>name</b>	<b>surname</b>	<b>country_name</b>
John	Smith	United States
Emma	Johnson	United Kingdom

- Queries to find passengers info with valid payment

```

1  SELECT personal_info.name, flights.flight_number, payment.total_amount
2  FROM Personal_info
3  INNER JOIN Passengers ON personal_info.pi_ID = passengers.pi_ID
4  INNER JOIN Flights ON passengers.ac_ID = flights.ac_ID AND passengers.rt_ID = flights.rt_ID
5  INNER JOIN payment ON passengers.payment_id = payment.payment_ID
6  WHERE payment.payment_status = true

```

<b>name</b>	<b>flight_number</b>	<b>total_amount</b>
John	AA101	364.50
Emma	BA202	380.50
Yuki	JL505	464.90
Sarah	QF606	950.25
Giovanni	AZ808	180.00
Maria	IB909	210.50

- Queries to sort payment by descending order

```

1  SELECT pay.payment_ID,pi.name,pay.total_amount
2  FROM payment pay
3  INNER JOIN Passengers p using(ps_id)
4  INNER JOIN Personal_info pi using(pi_id)
5  ORDER BY pay.total_amount DESC;

```

<b>payment_id</b>	<b>name</b>	<b>total_amount</b>
2006	Sarah	950.25
2005	Yuki	464.90
2002	Emma	380.50
2001	John	364.50
2007	David	288.00
2009	Maria	210.50
2008	Giovanni	180.00
2003	Pierre	82.02

- Queries to select expensive flights (above average)

```

1  SELECT flight_number,pay.total_amount
2  FROM Flights f
3  INNER JOIN payment pay ON f.ac_ID = pay.ac_ID AND f.rt_ID = pay.rt_ID
4  WHERE pay.total_amount > (
5    | |   SELECT AVG(total_amount) FROM payment
6  );

```

<b>flight_number</b>	<b>total_amount</b>
BA202	380.50
QF606	950.25
JL505	464.90
AA101	364.50

- Queries to select 3 most recent books

```

1  SELECT pay.payment_ID,pi.name,pi.surname,f.flight_number,pay.transaction_date,pay.total_amount
2  FROM payment pay
3  INNER JOIN Passengers p ON pay.ps_ID = p.ps_ID
4  INNER JOIN Personal_info pi ON p.pi_ID = pi.pi_ID
5  INNER JOIN Flights f ON pay.ac_ID = f.ac_ID AND pay.rt_ID = f.rt_ID
6  ORDER BY pay.transaction_date DESC
7  LIMIT 3;

```

<b>payment_id</b>	<b>name</b>	<b>surname</b>	<b>flight_number</b>	<b>transaction_d</b>	<b>total_amount</b>
2009	Maria	Garcia	IB909	2024-03-09	210.50
2008	Giovanni	Rossi	AZ808	2024-03-08	180.00
2007	David	Brown	AC707	2024-03-07	288.00

- Discount 10% to certain countries:

```

1  ALTER TABLE payment ADD COLUMN discount DECIMAL(5,2) DEFAULT 0;
2
3  UPDATE payment
4  SET total_amount = total_amount * 0.9,
5  discount=10.00
6  WHERE payment_ID IN (
7    |   SELECT p.payment_ID
8    |   FROM payment p
9    |   INNER JOIN Flights f USING (ac_id, rt_id)
10   |   INNER JOIN Route r USING (rt_id)
11   |   INNER JOIN Countries c USING (ct_id)
12   |   WHERE c.country_name = 'United States' or c.country_name='Germany' or c.country_name='France'
13 );
14
15  select * from payment;

```

payment_id	transaction_date	ps_id	ac_id	rt_id	total_amount	payment_status	discount
2002	2024-03-02	3002	502	102	380.50	true	0.00
2006	2024-03-06	3006	506	106	950.25	true	0.00
2008	2024-03-08	3008	508	108	180.00	true	0.00
2009	2024-03-09	3009	509	109	210.50	true	0.00
2005	2024-03-05	3005	505	105	464.90	true	10.00
2001	2024-03-01	3001	501	101	364.50	true	10.00
2004	2024-03-04	3004	504	104	171.68	true	10.00
2003	2024-03-03	3003	503	103	82.02	false	0.00
2007	2024-03-07	3007	507	107	288.00	false	0.00

- Selecting flights with certain date and duration:

The screenshot shows a SQL query editor interface. The query is:

```
SELECT * FROM flights
WHERE DATE(departure_time) = '2024-03-15'
  and duration = 420;
;
```

The results table has the following data:

ac_id	rt_id	flight_number	departure_time	duration
501	101	AA101	2024-03-15	420

- Queries to select money made from flights

```
1  SELECT SUM(total_amount) as money_made
2  FROM payment
3  ;
4
5  
```

Results    Chart    Export ▾

money_made
3092.35