

دانشگاه بین‌المللی امام خمینی



IMAM KHOMEINI
INTERNATIONAL UNIVERSITY

دانشگاه بین‌المللی امام خمینی (ره)

مهندسی کامپیوتر - دانشکده فنی مهندسی

توضیحات مربوط به دیتابیس و نمودار ارتباط موجودیت‌ها

(Database & ERD Documentation)

اعضای گروه:

حامد نظریان - امیرحسین موسوی نسب - امیرحسین ایزدی - آتنا باقری

گردآورنده:

سید امیرحسین موسوی نسب

نام گروه:

گروه E

استاد درس:

دکتر نستوه طاهری جوان

تابستان ۱۴۰۱

فهرست مطالب:

۰- مقدمه	3
۱- تحلیل سیستم	3
۲- انتخاب محیط مناسب برای طراحی نمودار ER	3
۳- ایجاد نمودار ارتباط موجودیت ها (ERD)	4
۴- بررسی اجزای نمودار ERD	4
• موجودیت Airplane	6
• موجودیت Airline	7
• موجودیت Flight	8
• موجودیت Passenger	10
• موجودیت Tour	11
۵- ارتباط میان موجودیت ها	13
• موجودیت های Airline و Flight	13
• موجودیت های Airplane و Flight	13
• موجودیت های Flight و Passenger	14
• موجودیت های Flight و Tour	15
۶- پیاده سازی پایگاه داده با استفاده از SQLite3	16
۷- جمع بندی نهایی	18

مقدمه:

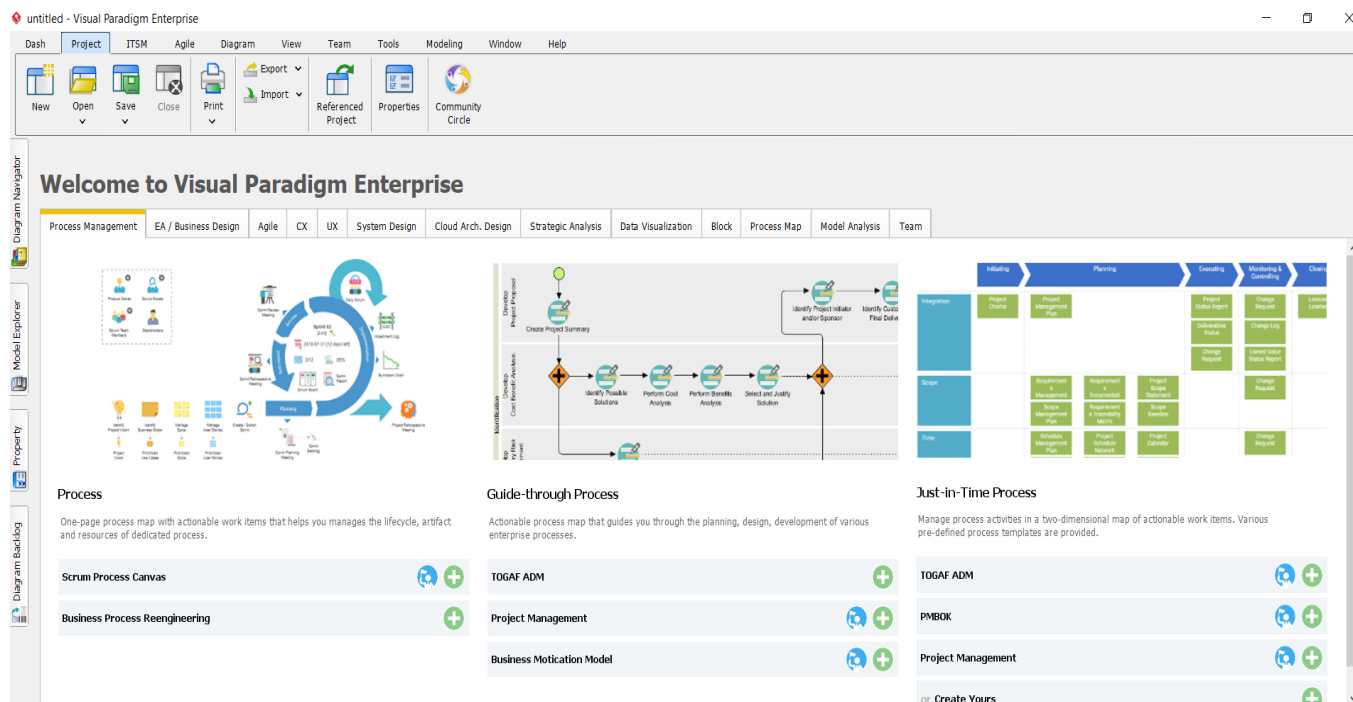
در این document قصد داریم تا ابتدا به بررسی کامل نمودار ارتباط موجودیت ها (Entity Relationship Diagram) یا به اختصار ERD میپردازیم. در این document، موجودیت ها و ارتباط میان آن ها و همچنین صفات قرار گرفته شده در هر موجودیت از نظر نوع داده و موارد مربوط به پیاده سازی database، مورد بررسی قرار می گیرند. در ادامه و پس از بررسی کامل نمودار ERD، به بررسی پیاده سازی پایگاه داده خواهیم پرداخت.

1. تحلیل سیستم

در ابتدا برای ایجاد یک نمودار ER درست، لازم بود تا سیستم مورد بحث به خوبی مورد تحلیل و بررسی قرار گیرد و عناصر موجود در این سیستم شناسایی شود. در ادامه لازم بود تا نیاز این عناصر به یکدیگر مورد بررسی قرار گیرد تا به ارتباط میان این عناصر پی ببریم و برای طراحی سیستم این ارتباطات را مد نظر قرار دهیم. در آخر و پس از بررسی های انجام شده، 5 موجودیت Airplane، Airline، Flight، Passenger و Tour انتخاب و صفات هر یک از این موجودیت ها شناسایی و مشخص گردید.

2. انتخاب محیط مناسب برای طراحی نمودار ER

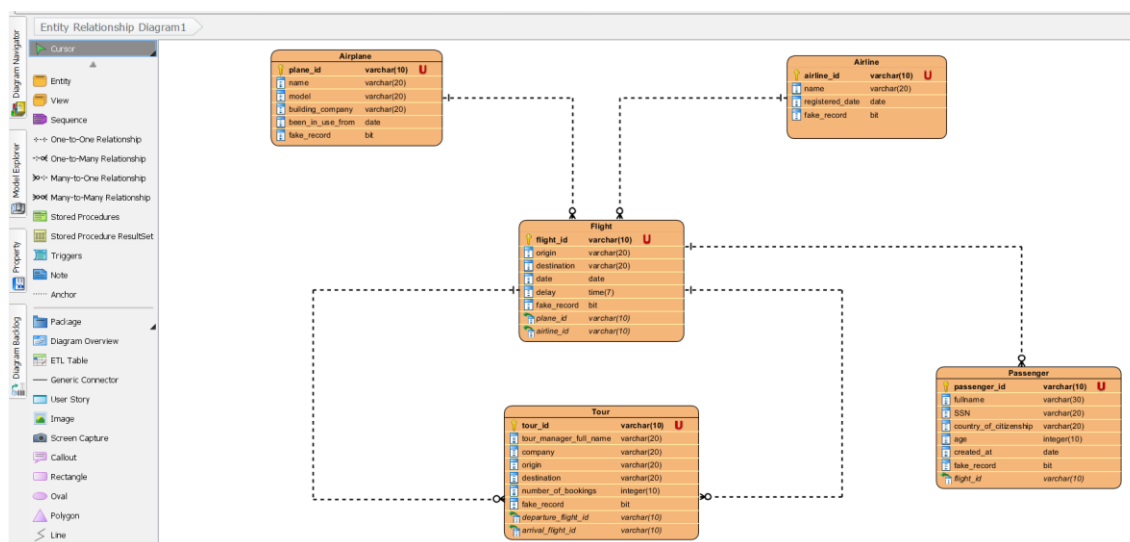
پس از تحلیل سیستم، نیاز به یک محیط گرافیکی خوب برای طراحی هر چه بهتر نمودار ارتباط موجودیت ها (ERD) داشتیم. پس از بررسی های انجام شده تصمیم به استفاده از نرم افزار Visual Paradigm گرفته شد که ابزاری قدرتمند برای تحلیل و طراحی سیستم می باشد. شکل 2.1، صفحه ابتدایی این نرم افزار قدرتمند را نمایش می دهد که از امکانات بسیاری برای تحلیل و طراحی سیستم های کامپیوتری برخوردار است.



شکل 2.1: محیط نرم افزار Visual Paradigm

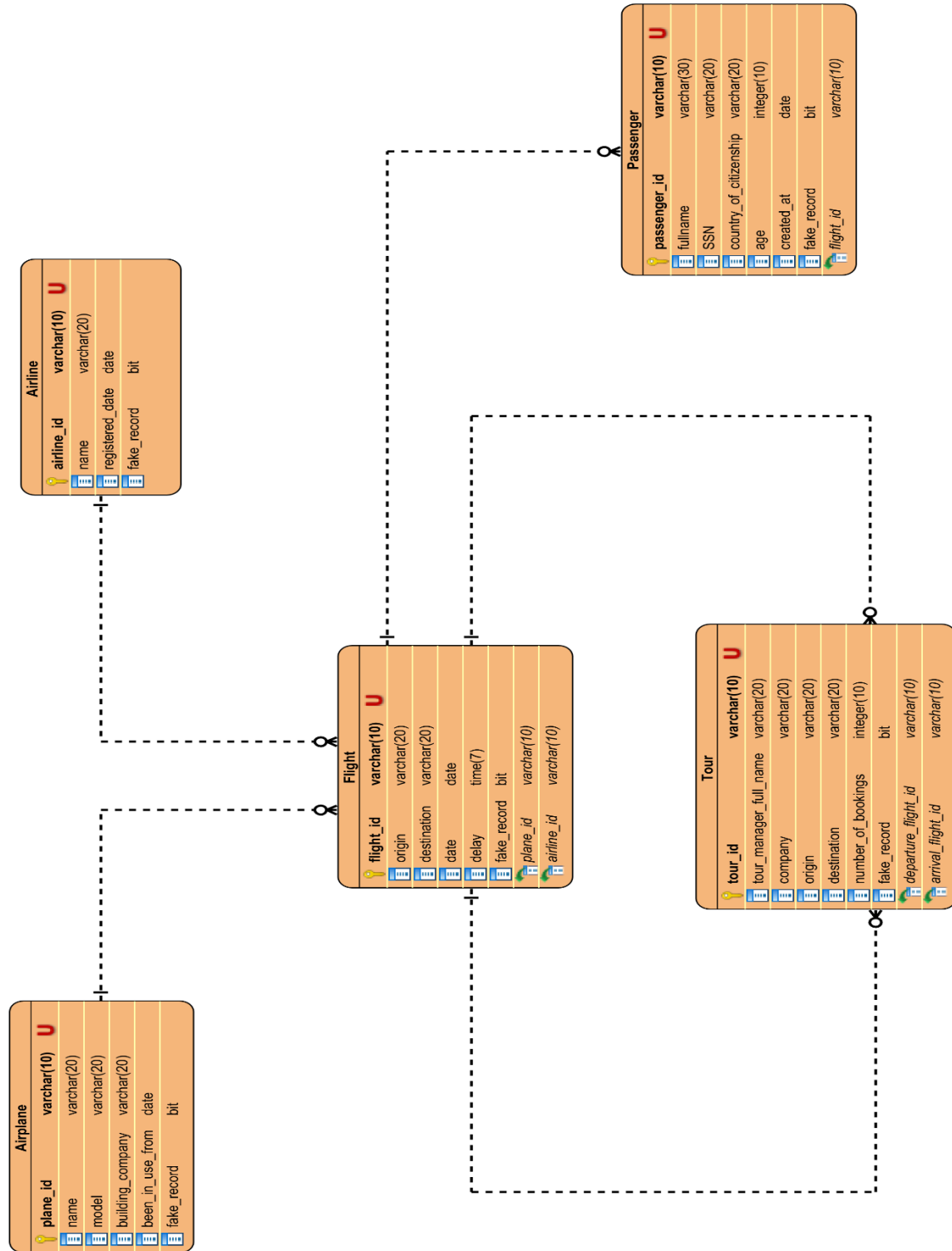
3. ایجاد نمودار ارتباط موجودیت ها (ERD)

پس از تحلیل سیستم و انتخاب محیط مناسب برای ایجاد نمودار، اقدام به ایجاد نمودار ارتباط موجودیت ها (ERD) گردید.



شکل 3.1: نمودار ERD طراحی شده در محیط نرم افزار Visual Paradigm

در ادامه نمودار ERD طراحی شده برای پروژه آمده است:



شکل 3.2: نمودار ERD پروژه







*** عکس با کیفیت و فایل نمودار ERD، در پوشه ERD در پروژه قابل مشاهده می باشد.

4. بررسی اجزای نمودار ERD

نمودار ERD طراحی شده دارای 5 جدول می باشد. این جداول عبارت اند از: Airplane، Airline، Flight، Passenger و Tour. در ادامه به بررسی هر کدام از این جداول و صفات آن ها می پردازیم.

*** توضیحات تکمیلی در مورد پیاده سازی پایگاه داده، در ادامه این document و پس از توضیحات مربوط به ERD آمده است که می توانید مشاهده بفرمایید.

• موجودیت Airplane





Airplane		
	plane_id	varchar(10) U
	name	varchar(20)
	model	varchar(20)
	building_company	varchar(20)
	been_in_use_from	date
	fake_record	bit

این موجودیت در واقع حاوی اطلاعات هواپیماهای موجود در فرودگاه می باشد. این جدول از 6 صفت با نام های plane_id، name، model، building_company، been_in_use_from و fake_record می باشد. در این جدول plane_id، کلید اصلی بوده و unique می باشد. همچنین سایر فیلدها نیز، NOT NULL هستند و حتما باید دارای مقدار باشند.

توضیحات مربوط به هر صفت در ادامه آمده است:

1. **Plane_id**: این فیلد بیانگر یک عبارت خاص به عنوان id برای هر هواپیماست و از نوع varchar و با محدودیت سائز 20 کاراکتر می باشد.
2. **name**: این فیلد مربوط به نام هر هواپیماست و حداکثر 20 کاراکتر می پذیرد.
3. **building_company**: این فیلد مربوط به کمپانی سازنده هر هواپیماست و از نوع varchar بوده و دارای محدودیت کاراکتری 20 می باشد.
4. **been_in_use_from**: این فیلد نشان می دهد که هر هواپیما از چه تاریخی مورد استفاده قرار گرفته است. نوع داده این فیلد از نوع date می باشد.
5. **fake_record**: این فیلد از نوع بیت بوده و نشان می دهد که آیا این داده از کاربر گرفته شده یا از داده های اولیه ای که توسط ما به دیتابیس داده شده، می باشد.

• موجودیت Airline









Airline		
	airline_id	varchar(10) U
	name	varchar(20)
	registered_date	date
	fake_record	bit

این موجودیت در واقع حاوی اطلاعات شرکت های هواپیمایی داخل فرودگاه که از آن ها بلیت تهیه می شود، است. این جدول از 4 صفت با نام های **airline_id** و **name** و **registered_date** و **fake_record** تهیه شده است. **airline_id**، کلید اصلی این جدول و **unique** می باشد. همچنین سایر فیلد های جدول نیز **NOT NULL** هستند و حتما باید دارای مقدار باشند.

توضیحات مربوط به هر صفت در ادامه آمده است:

1. **airline_id**: این فیلد نشان دهنده یک عبارت خاص 10 کاراکتری است که به هر airline نسبت داده می شود و آن را از airline های دیگر متمایز میکند.
2. **name**: این فیلد نشان دهنده نام هر airline می باشد و از نوع varchar و دارای محدودیت کاراکتری 20 می باشد.
3. **registered_date**: این فیلد نشان می دهد که هر شرکت هواپیمایی در چه تاریخی ثبت شده است به همین دلیل نوع داده این فیلد date می باشد.
4. **fake_record**: این فیلد از نوع بیت بوده و نشان می دهد که آیا این داده از کاربر گرفته شده یا از داده های اولیه ای که توسط ما به دیتابیس داده شده، می باشد

• موجودیت Flight









Flight		
	flight_id	varchar(10) U
	origin	varchar(20)
	destination	varchar(20)
	date	date
	delay	time(7)
	fake_record	bit
	plane_id	varchar(10)
	airline_id	varchar(10)

این موجودیت حاوی اطلاعات پرواز های فرودگاه می باشد. این جدول از 8 با نام های flight_id، origin، destination، date، delay، fake_record، plane_id و airline_id می باشد. همچنین flight_id نیز در این جدول کلید اصلی و دارای مقادیر unique می باشد. صفات plane_id و airline_id نیز کلید خارجی از جداول Airplane و Airline می باشند. تمام فیلد های جدول NOT NULL هستند و حتما باید دارای مقدار باشند.

توضیحات مربوط به هر صفت در ادامه آمده است:

1. **flight_id**: این فیلد نشان دهنده شماره هر پرواز 10 کاراکتری هر پرواز می باشد.
نوع داده این فیلد از نوع **varchar** می باشد.
2. **origin**: این فیلد نشان دهنده مبدا هر پرواز می باشد و از نوع **varchar** و دارای محدودیت 20 کاراکتری می باشد.
3. **destination**: این فیلد نشان دهنده مقصد هر پرواز می باشد و از نوع **varchar** و دارای محدودیت 20 کاراکتری می باشد.
4. **date**: این فیلد تاریخ انجام هر پرواز را نشان می دهد که دارای روز و ماه و سال می باشد. از همین رو نوع داده این فیلد **date** می باشد.
5. **delay**: این فیلد نشان دهنده تاخیر هر پرواز، و از نوع **time** می باشد.
6. **fake_record**: این فیلد از نوع بیت بوده و نشان می دهد که آیا این داده از کاربر گرفته شده یا از داده های اولیه ای که توسط ما به دیتابیس داده شده، می باشد.
7. **plane_id**: این فیلد که کلید خارجی از جدول **Airplane** می باشد، نشان دهنده id هواپیمای هر پرواز می باشد و نوع داده ای مشابه **plane_id** در جدول **Airplane** دارد.
8. **airline_id**: این فیلد که کلید خارجی از جدول **Airline** می باشد، نشان دهنده id هواپیمایی هر پرواز می باشد و نوع داده ای مشابه **airline_id** در جدول **Airline** دارد.

• موجودیت Passenger

Passenger		
	passenger_id	varchar(10) U
	full_name	varchar(30)
	SSN	varchar(20)
	country_of_citizenship	varchar(20)
	age	integer(10)
	created_at	date
	fake_record	bit
	flight_id	varchar(10)

این موجودیت دارای اطلاعات مسافرانی است که بلیت تهیه کرده اند. این جدول از 8 صفت با نام های passenger_id، full_name، SSN، country_of_citizenship، age، created_at، fake_record و flight_id می باشد. Passenger_id، کلید اصلی می باشد و تنها مقادیر unique را می پذیرد. فیلد flight_id نیز کلید خارجی از جدول Flight می باشد. همچنین تمام صفات این جدول NOT NULL هستند یعنی حتما باید مقدار داشته باشند.

توضیحات مربوط به هر صفت در ادامه آمده است:

1. **passenger_id**: این فیلد یک رشته کاراکتری 10 رقمی می باشد که برای هر

مسافر به صورت یکتاست. نوع داده این فیلد varchar و با محدودیت 10 کاراکتر می باشد.

2. **full_name**: این فیلد مشخصات نام و نام خانوادگی فرد را مشخص می کند. این

فیلد دارای محدودیت کاراکتر 30 تایی و از نوع varchar می باشد.

3. **SSN**: این فیلد کد شناسایی فرد است و دارای محدودیت کاراکتری 20 تایی و از

نوع varchar می باشد.

4. **country_of_citizenship**: این فیلد نشان دهنده کشور محل اقامت مسافر می

باشد و دارای محدودیت 20 کاراکتری و از نوع varchar می باشد.

5. **created_at**: این فیلد از نوع **date** می باشد و نشان دهنده تاریخ ثبت نام این مسافر می باشد.








6. **age**: این فیلد نشان دهنده سن فرد می باشد و از نوع **integer** می باشد.

7. **fake_record**: این فیلد از نوع بیت بوده و نشان می دهد که آیا این داده از کاربر گرفته شده یا از داده های اولیه ای که توسط ما به دیتابیس داده شده، می باشد.

8. **flight_id**: این فیلد کلید خارجی از جدول **Flight** بوده و نشان دهنده ی شماره پرواز هر شخص می باشد و نوع داده آن همانند **flight_id** در جدول **Flight** می باشد.

9.

• موجودیت Tour

Tour		
 tour_id	varchar(10)	U
 tour_manager_full_name	varchar(20)	
 company	varchar(20)	
 origin	varchar(20)	
 destination	varchar(20)	
 number_of_bookings	integer(10)	
 fake_record	bit	
 departure_flight_id	varchar(10)	
 arrival_flight_id	varchar(10)	

این موجودیت، اطلاعات تور های موجودیت برای ارائه به مسافران و شرکت های مسافرتی را می دهد. این موجودیت دارای 9 صفت با نام های **tour_id**، **company**، **origin**، **tour_manager_full_name**، **destination**، **number_of_booking**، **fake_record**، **departure_flight_id** و **arrival_flight_id** می باشد. صفت **tour_id**، کلید اصلی و دارای مقدار **unique** می باشد. صفات **departure_flight_id** و **arrival_flight_id** نیز کلید های خارجی و هر دو از جدول **Flight** می باشند. همچنین تمام صفات این جدول **NOT NULL** هستند یعنی حتما باید مقدار داشته باشند.

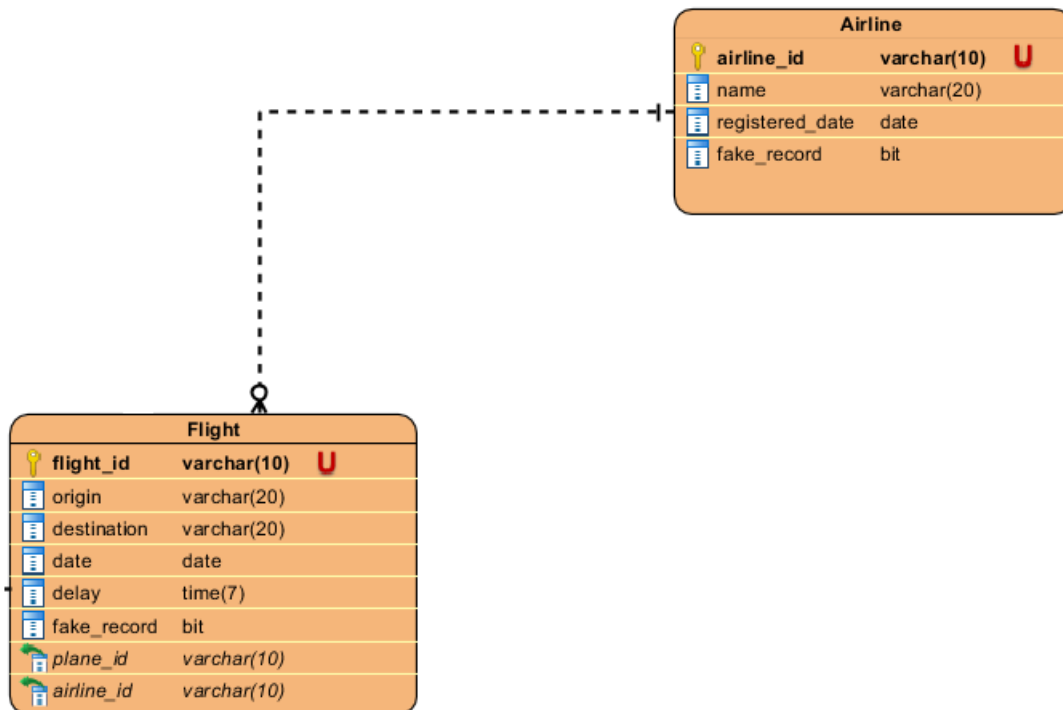
توضیحات مربوط به هر صفت در ادامه آمده است:

1. **tour_id**: این فیلد در واقع id مخصوص به هر تور را مشخص می کند و از نوع varchar و با محدودیت 10 کاراکتری می باشد.
2. **tour_manager_full_name**: این فیلد نشان دهنده ی نام کامل مدیر تور می باشد و از نوع varchar و دارای محدودیت 20 کاراکتری می باشد.
3. **company**: این فیلد نشان دهنده ی شرکتی ست که تور را ارائه می کند و از نوع داده varchar با محدودیت 20 کاراکتر می باشد.
4. **origin**: این فیلد نشان دهنده ی مبدا تور می باشد و از نوع داده varchar و دارای محدودیت 20 کاراکتری می باشد.
5. **destination**: این فیلد نشان دهنده ی مقصد تور می باشد و از نوع داده varchar و دارای محدودیت 20 کاراکتری می باشد.
6. **nember_of_booking**: این فیلد نشان دهنده ی تعداد بلیت های رزرو شده توسط این تور می باشد و از همین رو از نوع integer در نظر گرفته شده است.
7. **fake_record**: این فیلد از نوع بیت بوده و نشان می دهد که آیا این داده از کاربر گرفته شده یا از داده های اولیه ای که توسط ما به دیتابیس داده شده، می باشد
8. **departure_flight_id**: این فیلد، کلید خارجی از جدول Flight و نشان دهنده ی id پرواز مبدا می باشد. نوع داده این فیلد دقیقا همان نوع داده Flight_id در جدول Flight می باشد.
9. **arrival_flight_id**: این فیلد، کلید خارجی از جدول Flight و نشان دهنده ی id پرواز مبدا می باشد. نوع داده این فیلد دقیقا همان نوع داده Flight_id در جدول Flight می باشد.

5. ارتباط میان موجودیت ها

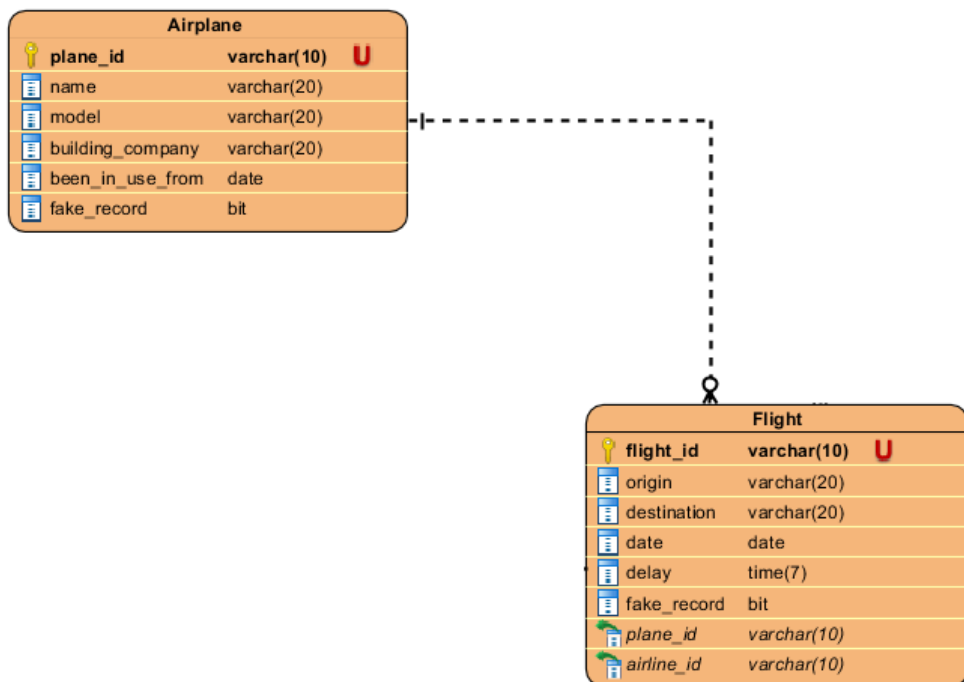
در این قسمت به ارتباط میان موجودیت های مختلف موجود در پایگاه داده می پردازیم.

• موجودیت های Airline و Flight



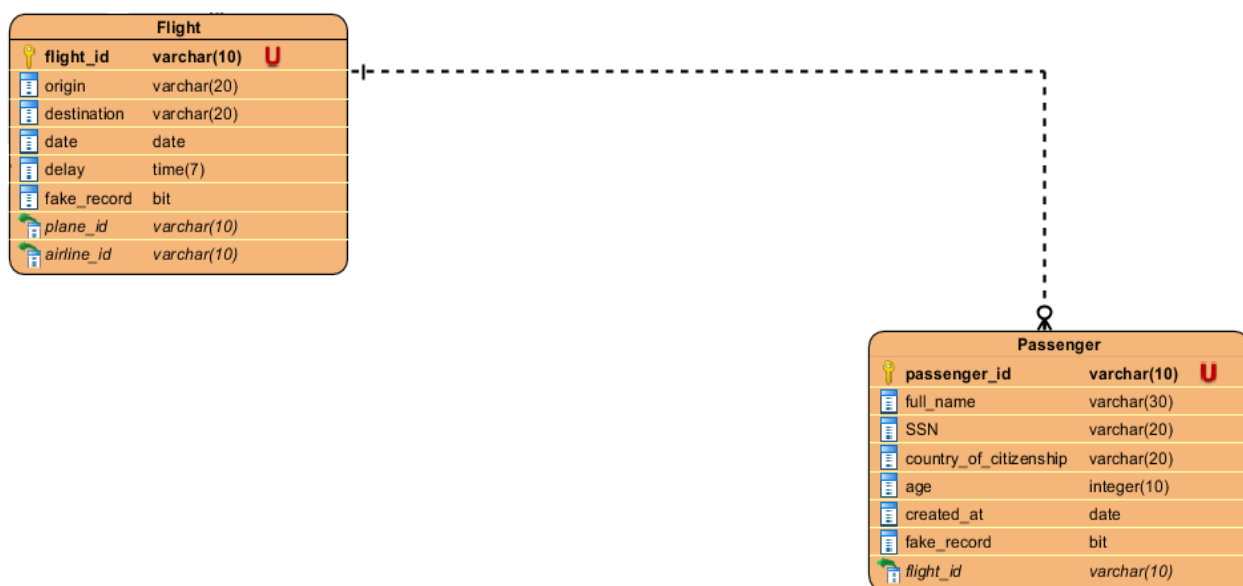
همانطور که مشاهده می کنید دو موجودیت **Airline** و **Flight**، رابطه ای از نوع **One to Many** دارند. به این صورت که هر پرواز به یک هواپیمایی تعلق دارد اما هر هواپیمایی می تواند چندین پرواز داشته باشد.

• موجودیت های Airplane و Flight



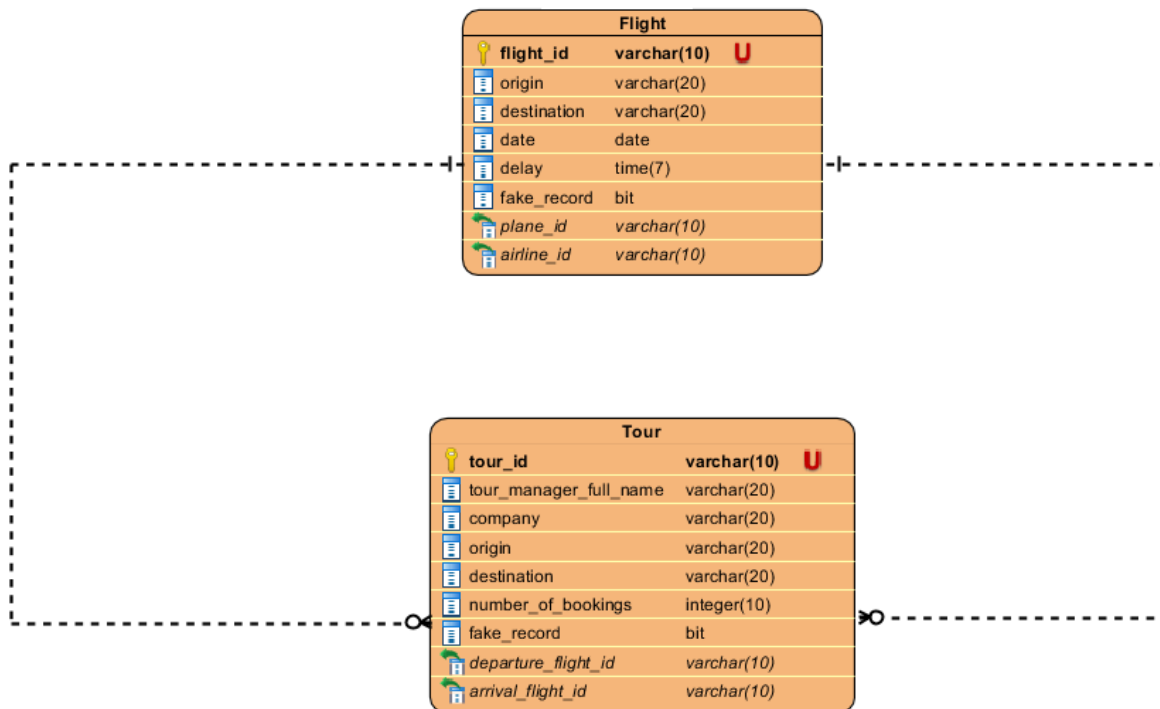
همانطور که مشاهده میکنید دو موجودیت Airplane و Flight، رابطه One to Many دارند. به این صورت که هر پرواز متعلق به یک هواپیما می باشد اما هر هواپیما می تواند پرواز های مختلفی در زمان های متفاوت داشته باشد.

• موجودیت های Flight و Passenger



همانطور که مشاهده می کنید دو موجودیت Flight و Passenger با هم رابطه One to Many دارند. به این صورت که هر مسافر تنها متعلق به یک پرواز است اما هر پرواز چندین مسافر دارد.

• موجودیت های Flight و Tour



همانطور که مشاهده می کنید دو موجودیت Flight و Tour با هم رابطه دو رابطه One to Many دارند. به این صورت که هر تور یک پرواز رفت و یک پرواز برگشت دارد. اما هر پرواز میتواند متعلق به چندین تور باشد.

6. پیاده سازی پایگاه داده با استفاده از SQLite3

در ابتدا برای پیاده سازی این پایگاه داده از MS SQL Server استفاده شده بود اما به دلیل مشکلات به وجود آمده در بحث Config و عدم انطباق مناسب MS SQL Server با پایتون و پس از مشورت سرگروه با استاد، مقرر گردید تا از SQLite3 که انطباق پذیری بالایی با زبان پایتون دارد برای پیاده سازی پایگاه داده استفاده شود. پیاده سازی این پایگاه داده تماماً به وسیله کد و با زدن Query های مربوطه انجام گردید. کد های مربوط به پیاده سازی پایگاه داده در ادامه آمده است. همچنین فایل پایتون سازنده پایگاه داده در قالب فایل Airport_Database_Creator.py موجود است و می توانید مشاهده بفرمایید.

```
1  import sqlite3
2
3  conn = sqlite3.connect(r'./Airport.db')
4  conn.execute("PRAGMA foreign_keys = 1")
5  cur = conn.cursor()
6  cur.execute("""CREATE TABLE IF NOT EXISTS Airplane (
7                  plane_id          VARCHAR (10) PRIMARY KEY
8                                  UNIQUE
9                                  NOT NULL,
10                 name              VARCHAR (20) NOT NULL,
11                 model              VARCHAR (20) NOT NULL,
12                 building_company   VARCHAR (20) NOT NULL,
13                 been_in_use_from   DATE        NOT NULL,
14                 fake_record        BOOLEAN     NOT NULL
15             );
16 """)
17 conn.commit()
18 cur.execute("""CREATE TABLE IF NOT EXISTS Airline (
19                 airline_id         VARCHAR (10) PRIMARY KEY
20                                 NOT NULL
21                                 UNIQUE,
22                 name               VARCHAR (20) NOT NULL,
23                 registered_date    DATE        NOT NULL,
24                 fake_record        BOOLEAN     NOT NULL
25             );
26 """)
27 conn.commit()
```


این تصویر ساخت database و جداول Airplane و Airline را نشان می دهد.

```
28 cur.execute("""CREATE TABLE IF NOT EXISTS Flight (
29     flight_id    VARCHAR (10) PRIMARY KEY
30                 UNIQUE
31                 NOT NULL,
32     origin       VARCHAR (10) NOT NULL,
33     destination  VARCHAR (10) NOT NULL,
34     date         DATE         NOT NULL,
35     delay        TIME (7)     NOT NULL,
36     plane_id     VARCHAR (10) NOT NULL,
37     airline_id   VARCHAR (10) NOT NULL,
38     fake_record  BOOLEAN (10) NOT NULL,
39     FOREIGN KEY (
40         plane_id
41     )
42     REFERENCES Airplane (plane_id),
43     FOREIGN KEY (
44         airline_id
45     )
46     REFERENCES Airline (airline_id)
47 );
48 """)
49 conn.commit()
```

این تصویر Query زده شده برای ساخت جدول Flight را نمایش می دهد.

```
50 cur.execute("""CREATE TABLE IF NOT EXISTS Passenger (
51     passenger_id    VARCHAR (20) PRIMARY KEY
52                     UNIQUE
53                     NOT NULL,
54     flight_id       VARCHAR (10) NOT NULL,
55     SSN             VARCHAR (20) NOT NULL,
56     country_of_citizenship VARCHAR (20) NOT NULL,
57     fake_record     BOOLEAN     NOT NULL,
58     age             INT         NOT NULL,
59     created_at      DATE        NOT NULL,
60     fullname        VARCHAR (30) NOT NULL,
61     FOREIGN KEY (
62         flight_id
63     )
64     REFERENCES Flight (flight_id)
65 );
66 """)
67 conn.commit()
```

این تصویر Query زده شده برای ساخت جدول Passenger را نشان می دهد.

```
68 cur.execute("""CREATE TABLE IF NOT EXISTS Tour (
69     tour_id          VARCHAR (10) PRIMARY KEY
70                     NOT NULL
71                     UNIQUE,
72     tour_manager_full_name VARCHAR (30) NOT NULL,
73     company           VARCHAR (20) NOT NULL,
74     origin            VARCHAR (20) NOT NULL,
75     destination       VARCHAR (20) NOT NULL,
76     number_of_booking INT          NOT NULL,
77     departure_flight_id VARCHAR (10) NOT NULL,
78     arrival_flight_id  VARCHAR (10) NOT NULL,
79     fake_record        BOOLEAN     NOT NULL,
80     FOREIGN KEY (
81         departure_flight_id
82     )
83     REFERENCES Flight (flight_id),
84     FOREIGN KEY (
85         arrival_flight_id
86     )
87     REFERENCES Flight (flight_id)
88 );
89 """)
90 conn.commit()
```

و در آخر نیز کد مربوط به ساخت جدول Tour را مشاهده می فرمایید.

7. جمع بندی نهایی

در این پیاده سازی پایگاه داده این پروژه سعی شد تا حد امکان از مطالب گفته شده در درس پایگاه داده استفاده شده و طراحی و تغییرات در دیتابیس به وسیله Query و با شیوه درست و اصولی صورت گیرد. فایل Database نهایی در قالب فایل Airport.db در پروژه استفاده شده است.