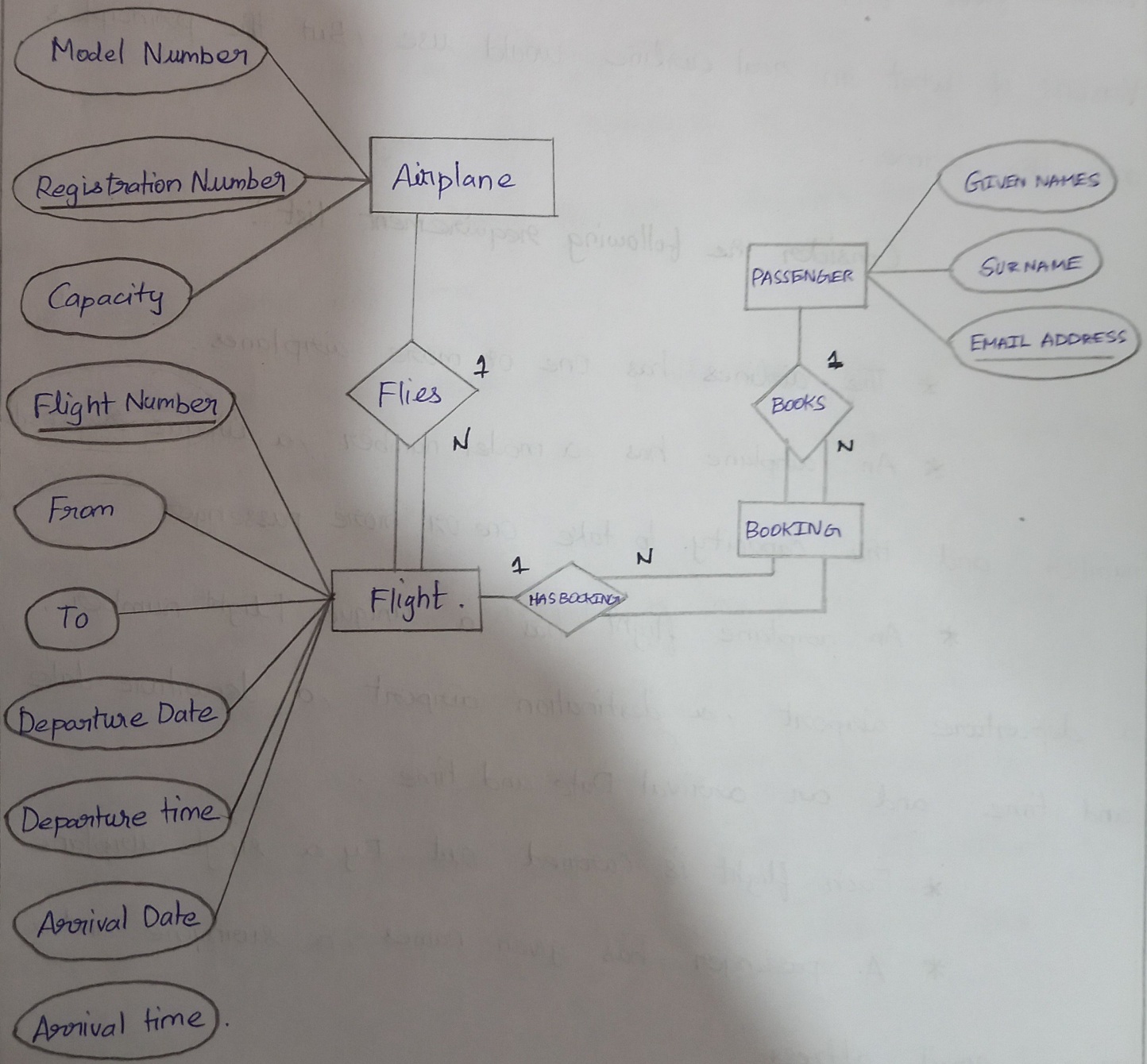
1. DRAW AN ER DIAGRAM FOR THE FLIGHT DATABASE WITH ALL COMPONENTS AND Explain:

The Flight Database stores details about an airlines fleet, flights and seat Bookings. Again its hugely simplified Versions of what an real airline would use, But the principles are the same.

Consider the following requirement list:

* The airlines has one or more airplanes
* An airplane has a model number, a unique registration number, and the capacity to take one or more passenger.
* An airplane flight has a unique Flight number, a departure airport, a destination airport, a departure date and time and an arrival Date and time.
* Each flight is carried out By a single airplane
* A passenger has given names, a surname and a unique email address.
* A passenger Can Book a seat on a flight.

The ER Diagram Derived from our requirement is shown in figure.



* An airplane is unique identified By its Registration Number , so we can use this as the primary key .

* An Flight is uniquely identified By its Flight Number , so we can use the Flight Number as primary key , The Departure and destination airports are captured in the Form and to attributes , and we have separate attributes for the Departure and arrival date and time .

* Because no two passengers will share an email address , we can use the email address as the primary key For the Passenger entity .

* An Airplane Can be Involved in any number of flights . While each flights uses exactly one airplanes , so the Flies Relationship Between the Airplane and flight relationship has Cardinality . 1 : N ; Because a flight Cannot exist without an airplane . The flight entity participates totally in this Relationship .

* A passenger can Book any number of flights, While a flight Can be Booked By any number passenger. As Discussed earlier in Intermediate Entities" we could Specify and M:N Books relationship Between the passenger and flight relationship. But Considering the issue more Carefully shows that there is a hidden entity here; the Booking itself. We Capture this By Creating the intermediate entity Booking and 1:N Relationship Between it and the Passenger and flight entities. Identifying such entities allows us to get Better picture of Requirements.

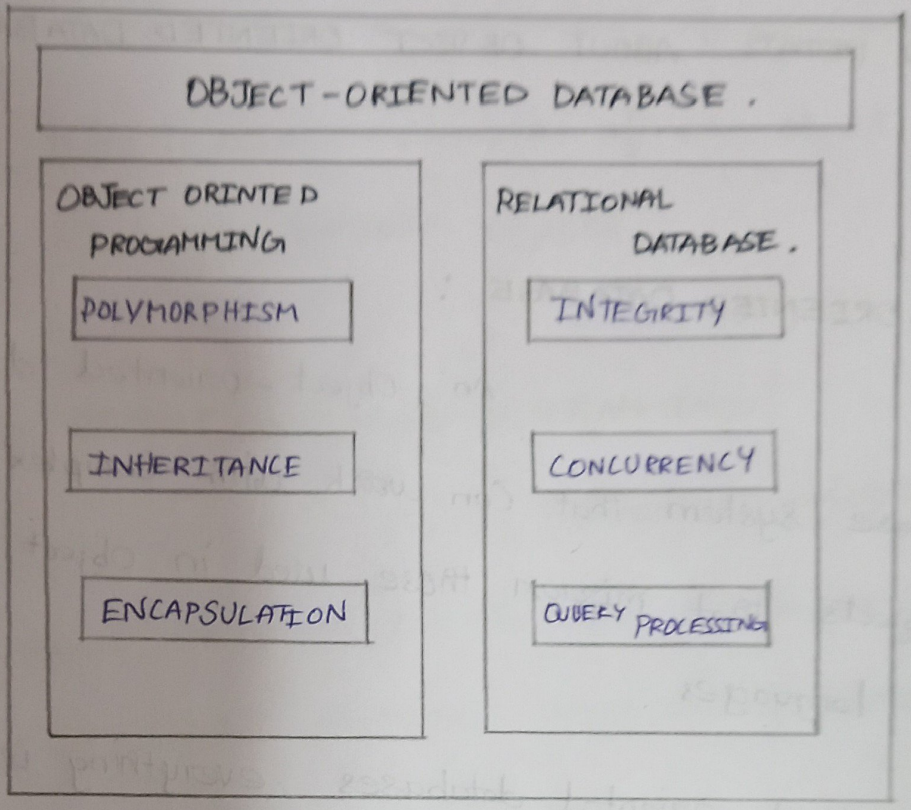
2. DISCUSS IN DETAIL ABOUT OBJECT ORIENTED DATABASE & XML DATABASE :

- OBJECT - ORIENTED DATABASE :

An Object - oriented database (OOD) is a Database system that can work with Complex data object - that is objects that mirror those used in object - oriented Programming languages.

In object - oriented databases, everything is a object, and many object are quite Complex having different properities and method. An object - oriented database management system works in Concert with an object - oriented programming language. to facilate the storage and retrieval of object oriented data.

* An object - oriented database is organised around objects rather than actions, and data rather than logic. For examples, a Multimedia record a relational Database can be a definable data object, as opposed to be a Alphanumberic Value.



* Object-oriented Database direcly deal with data as Complete object. All the Information Comes in one instantly avaiable object package instead of multiple tables.

• OBJECTS ; Are the Basic Building Block and an instance of a class, where the type is either Built-in or User-Defined.

• CLASSES : Are provide a schema or Blueprint for Object, Defining the Behaviour.

- Methods Determine the behaviour of a Class
- Pointers helps access elements of an object Database and establish relation between objects.

* Concepts of OBJECT ORIENTED PROGRAMMING:

Objects-oriented database closely relate to object-oriented programming concepts. The Four main ideas of object oriented programming are:

- POLYMORPHISM
- Inheritance
- Encapsulation.

* POLYMORPHISM:

Is the capability of an object to take multiple Forms. This ability allows the same program Code to work with different Data types. Both a Car and a bike are able to Break. But the mechanism is different. The action Break is a polymorphism. The defined action is polymorphic.

* Inheritance :

It Creates a hierarchial relationship between related classes while making parts of Code reusable. Defining new types inherits all the existing class fields and methods plus further extends them.

* Encapsulation :

Encapsulation is a ability to group data and mechanisms into a single object to provide access Protection. Through this process, pieces of information and Details of how an object works are hidden.

Resulting in data and function Security, classes interact with each other through methods without the need to know the particular methods Works.

A class encapsulates all the car Information into one entity, where some elements are modifiable while some are not.

* XML DATABASE :

XML Database is used to store Huge amount of Information in the XML Format. As the use of XML is increasing in every Field. it is required to have a secured place to store the XML Documents. The data stored in the DataBase Can be queried using XQuery. Serialized and exported into a desired Format.

XML DATABASES TYPES are two major types of XML Database.

- XML - enabled
- Native XML (NXD).

• XML - Enabled Databases :

XML enabled database is nothig but the extension provided For the Conversion of XML Document. This is relational Database, where data stored in tables Consisting of Rows and columns. The table Contains set of records, which in turn Consist of fields.

• NATIVE XML DATABASE :

Native XML DATABASE is Based on the Container rather than table format. It can store large amount of XML Docuement and dta. Native XML Database is queried By the X-path - expressions.

Native XML Database has an advantage over the XML enabled database. It is highly Capable to store, query and maintain the XML Docuement than XML-enabled database

Examples For XML DATABASE :

```
<?XML Version. = "1.0"?>
< Contact_info >
< Contact 1>
< name > tanmay Pathil </name>
< Company > Tutorialspoint </Company>
< phone > (011) 123-4567 </phone>
</Contact1>
<Contact 2>
<name > Manisha Patel </name>
< Company > tutorial Point </ Company>
<Phone >011) 7891-4567 </ Phone>
<Contact 2> .</Contact-info>
```