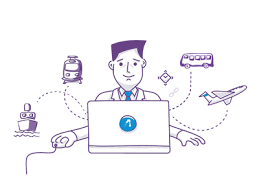
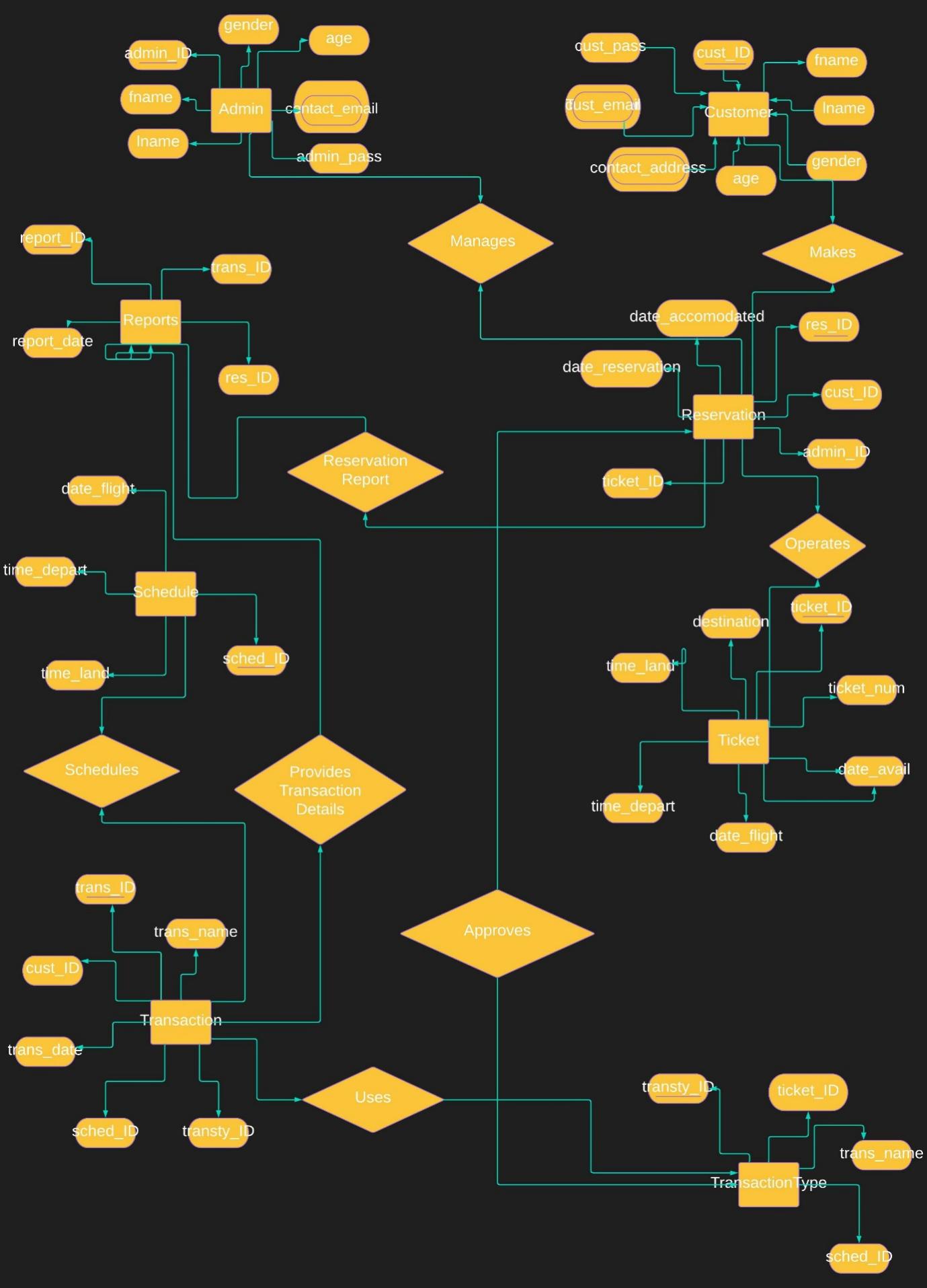
**AIRLINE TICKET RESERVATION MANAGEMENT SYSTEM**





|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| admin TABLE | |  |  |  |  |  |  |
| admin\_ID | fname | lname | gender | age | contact\_email | admin\_pass | |
| 10100 | Abdul | Aggarwal | M | 18 | [abdul@gmail.com](mailto:abdul@gmail.com) | a121010 |  |
| 10101 | Ishani | Gupta | F | 20 | [ishani@gmail.com](mailto:ishani@gmail.com) | b121011 |  |
| 10102 | Ajay | Fukta | M | 25 | [ajay@gmail.com](mailto:ajay@gmail.com) | c121012 |  |
| 10103 | Harshita | Goel | F | 29 | [harshita@gmail.com](mailto:harshita@gmail.com) | d121013 |  |
| 10104 | Gaurav | Garg | M | 32 | [gaurav@gmail.com](mailto:gaurav@gmail.com) | e121014 |  |
| 10105 | Garry | Balgurjot | M | 22 | [garry@gmail.com](mailto:garry@gmail.com) | f121015 |  |

**Functional Dependencies:**

**Admin:-**

contact\_email,admin\_pass-> fname,lname,gender,age

admin\_ID->contact\_email,fname,lname,gender,age

**NORMALISATION :**

**Admin:-**

Here,

Admin\_ID is Primary Key

admin\_ID is Candidate Key as It is traversing all other attributes.

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is not Normalised to 3rd Normal Form as condition/Functional Dependency= Candidate Key->NonPrime is there .

So we will Decompose the table into two tables.

Decomposed Table:

Table 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| fname | lname | gender | age | contact\_email | admin\_pass | |
| Abdul | Aggarwal | M | 18 | [abdul@gmail.com](mailto:abdul@gmail.com) | a121010 |  |
| Ishani | Gupta | F | 20 | [ishani@gmail.com](mailto:ishani@gmail.com) | b121011 |  |
| Ajay | Fukta | M | 25 | [ajay@gmail.com](mailto:ajay@gmail.com) | c121012 |  |
| Harshita | Goel | F | 29 | [harshita@gmail.com](mailto:harshita@gmail.com) | d121013 |  |
| Gaurav | Garg | M | 32 | [gaurav@gmail.com](mailto:gaurav@gmail.com) | e121014 |  |
| Garry | Balgurjot | M | 22 | [garry@gmail.com](mailto:garry@gmail.com) | f121015 |  |

Table 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| admin\_ID | fname | lname | gender | age | contact\_email |
| 10100 | Abdul | Aggarwal | M | 18 | [abdul@gmail.com](mailto:abdul@gmail.com) |
| 10101 | Ishani | Gupta | F | 20 | [ishani@gmail.com](mailto:ishani@gmail.com) |
| 10102 | Ajay | Fukta | M | 25 | [ajay@gmail.com](mailto:ajay@gmail.com) |
| 10103 | Harshita | Goel | F | 29 | [harshita@gmail.com](mailto:harshita@gmail.com) |
| 10104 | Gaurav | Garg | M | 32 | [gaurav@gmail.com](mailto:gaurav@gmail.com) |
| 10105 | Garry | Balgurjot | M | 22 | [garry@gmail.com](mailto:garry@gmail.com) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CUSTOMER TABLE | |  |  |  |  |  |  |
| cust\_ID | fname | lname | gender | age | contact\_add | cust\_email | cust\_pass |
| 11100 | Rahul | Sharma | M | 25 | New Delhi | [rahul@gmail.com](mailto:rahul@gmail.com) | 141010 |
| 11101 | Abhay | Deol | M | 22 | Mumbai | [abhay@gmail.com](mailto:abhay@gmail.com) | 141011 |
| 11102 | Priya | Sachdeva | F | 26 | Rajasthan | [priya@gmail.com](mailto:priya@gmail.com) | 141012 |
| 11103 | Raghav | Khurrana | M | 22 | Banglore | [raghav@gmail.com](mailto:raghav@gmail.com) | 141013 |
| 11104 | Shivani | Verma | F | 20 | Kerela | [shivani@gmail.com](mailto:shivani@gmail.com) | 141014 |
| 11105 | Diya | Sharma | F | 28 | Punjab | [diya@gmail.com](mailto:diya@gmail.com) | 141015 |

**Functional Dependencies:**

**Customer:-**

cust\_ID,cust\_pass -> fname,lname,gender,age,contact\_add,cust\_email

cust\_ID -> contact\_add,cust\_email

**NORMALISATION :**

**Customer:-**

Here,

cust\_ID is Primary Key

cust\_ID is Candidate Key as It is traversing all other attributes.

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is not Normalised to 3rd Normal Form as condition/Functional Dependency= Candidate Key->NonPrime is there .

So we will Decompose the table into two tables.

Decomposed Table:

Table 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | cust\_ID | fname | lname | gender | age | contact\_add | cust\_email | cust\_pass | | 11100 | Rahul | Sharma | M | 25 | New Delhi | [rahul@gmail.com](mailto:rahul@gmail.com) | 141010 | | 11101 | Abhay | Deol | M | 22 | Mumbai | [abhay@gmail.com](mailto:abhay@gmail.com) | 141011 | | 11102 | Priya | Sachdeva | F | 26 | Rajasthan | [priya@gmail.com](mailto:priya@gmail.com) | 141012 | | 11103 | Raghav | Khurrana | M | 22 | Banglore | [raghav@gmail.com](mailto:raghav@gmail.com) | 141013 | | 11104 | Shivani | Verma | F | 20 | Kerela | [shivani@gmail.com](mailto:shivani@gmail.com) | 141014 | | 11105 | Diya | Sharma | F | 28 | Punjab | [diya@gmail.com](mailto:diya@gmail.com) | 141015 | |
|  |
|  |
|  |
|  |
|  |

Table 2

|  |  |  |
| --- | --- | --- |
| contact\_add | cust\_email | cust\_ID |
| New Delhi | [rahul@gmail.com](mailto:rahul@gmail.com) | 11100 |
| Mumbai | [abhay@gmail.com](mailto:abhay@gmail.com) | 11101 |
| Rajasthan | [priya@gmail.com](mailto:priya@gmail.com) | 11102 |
| Banglore | [raghav@gmail.com](mailto:raghav@gmail.com) | 11103 |
| Kerela | [shivani@gmail.com](mailto:shivani@gmail.com) | 11104 |
| Punjab | [diya@gmail.com](mailto:diya@gmail.com) | 11105 |

|  |  |  |  |
| --- | --- | --- | --- |
| REPORTS TABLE | |  |  |
| report\_ID | trans\_ID | res\_ID | report\_date |
| 12100 | 17100 | 13100 | 21-10-2020 |
| 12101 | 17101 | 13101 | 01-11-2020 |
| 12102 | 17102 | 13102 | 02-11-2020 |
| 12103 | 17103 | 13103 | 09-11-2020 |
| 12104 | 17104 | 13104 | 12-12-2020 |
| 12105 | 17105 | 13105 | 03-01-2021 |

**Functional Dependencies:**

**Reports:-**

|  |  |
| --- | --- |
| report\_ID -> res\_ID,trans\_ID,report\_date | |
| res\_ID -> report\_date |  |

**NORMALISATION :**

**Reports:-**

Here,

report\_ID is Primary Key

report\_ID is candidate key

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is not Normalised to 3rd Normal Form as condition/Functional Dependency= Non Prime->NonPrime is there .

So we will Decompose the table into two tables.

Decomposed Table:

Table 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | report\_ID | trans\_ID | res\_ID | report\_date | | 12100 | 17100 | 13100 | 21-10-2020 | | 12101 | 17101 | 13101 | 01-11-2020 | | 12102 | 17102 | 13102 | 02-11-2020 | | 12103 | 17103 | 13103 | 09-11-2020 | | 12104 | 17104 | 13104 | 12-12-2020 | | 12105 | 17105 | 13105 | 03-01-2021 | |
|  |
|  |
|  |
|  |

Table 2

|  |  |
| --- | --- |
| res\_ID | report\_date |
| 13100 | 21-10-2020 |
| 13101 | 01-11-2020 |
| 13102 | 02-11-2020 |
| 13103 | 09-11-2020 |
| 13104 | 12-12-2020 |
| 13105 | 03-01-2021 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RESERVATION TABLE | | |  | |  | |
| res\_ID | cust\_ID | admin\_ID | | ticket\_ID | | date\_reserve | | date\_accom |
| 13100 | 11100 | 10100 | | 16100 | | 05-02-2020 | | 20-03-2020 |
| 13101 | 11101 | 10101 | | 16101 | | 15-02-2020 | | 21-03-2020 |
| 13102 | 11102 | 10102 | | 16102 | | 03-03-2020 | | 22-03-2020 |
| 13103 | 11103 | 10103 | | 16103 | | 22-03-2020 | | 23-03-2020 |
| 13104 | 11104 | 10104 | | 16104 | | 02-07-2020 | | 24-03-2020 |
| 13105 | 11105 | 10105 | | 16105 | | 09-12-2020 | | 25-03-2020 |

**Functional Dependencies:**

**Reservation:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| admin\_ID -> res\_ID,cust\_ID,ticket\_ID,date\_res,date\_accom | | | | |
| res\_ID -> res\_ID,cust\_ID,ticket\_ID,date\_reserve,date\_accom | | | | |
| cust\_ID -> ticket\_ID |  |  |  |  |

**NORMALISATION :**

**Reservation:-**

Here,

res\_ID is Primary Key

admin\_ID is candidate key

admin\_ID,cust\_ID,ticket\_ID is foreign key

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is not Normalised to 3rd Normal Form as condition/Functional Dependency= Non Prime->NonPrime is there .

So we will Decompose the table into two tables.

Decomposed Table:

Table 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | res\_ID | cust\_ID | admin\_ID | ticket\_ID | | 13100 | 11100 | 10100 | 16100 | | 13101 | 11101 | 10101 | 16101 | | 13102 | 11102 | 10102 | 16102 | | 13103 | 11103 | 10103 | 16103 | | 13104 | 11104 | 10104 | 16104 | | 13105 | 11105 | 10105 | 16105 | |
|  |
|  |
|  |
|  |

Table 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| res\_ID | cust\_ID | ticket\_ID | date\_reserve | date\_accom |
| 13100 | 11100 | 16100 | 05-02-2020 | 20-03-2020 |
| 13101 | 11101 | 16101 | 15-02-2020 | 21-03-2020 |
| 13102 | 11102 | 16102 | 03-03-2020 | 22-03-2020 |
| 13103 | 11103 | 16103 | 22-03-2020 | 23-03-2020 |
| 13104 | 11104 | 16104 | 02-07-2020 | 24-03-2020 |
| 13105 | 11105 | 16105 | 09-12-2020 | 25-03-2020 |
|  |  |  |  |  |

Table 3

|  |  |
| --- | --- |
| cust\_ID | ticket\_ID |
| 11100 | 16100 |
| 11101 | 16101 |
| 11102 | 16102 |
| 11103 | 16103 |
| 11104 | 16104 |
| 11105 | 16105 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SCHEDULE TABLE | | |  | |  | |
| S.No. | sched\_ID | date\_flight | | time\_departure | | time\_laNd | |
| 1 | 15100 | 21-10-2020 | | 15:30 | | 19:10 | |
| 2 | 15101 | 01-11-2020 | | 21:00 | | 23:45 | |
| 3 | 15102 | 02-11-2020 | | 19:00 | | 22:00 | |
| 4 | 15103 | 09-12-2020 | | 04:00 | | 07:00 | |
| 5 | 15104 | 12-12-2020 | | 06:00 | | 10:00 | |
| 6 | 15105 | 03-01-2021 | | 05:00 | | 07:30 | |

**Functional Dependencies:**

**Schedule:-**

|  |
| --- |
| sched\_ID -> date\_flight,time\_depart,time\_land |

**NORMALISATION :**

**Schedule:-**

Here,

sched\_ID is Primary Key

sched\_ID is candidate key

admin\_ID,cust\_ID,ticket\_ID is foreign key

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is already Normalised to 3rd Normal Form as condition/Functional Dependency= Prime->NonPrime is there also No Transitive Dependency.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ticket TABLE | | |  | |  | |
| ticket\_ID | ticket\_num | date\_avail | | date\_flight | | time\_depart | | time\_land | destination | |
| 16100 | 601 | 30-10-2020 | | 31-10-2020 | | 15:30 | | 19:10 | America |  |
| 16101 | 602 | 16-02-2021 | | 07-02-2021 | | 21:00 | | 23:45 | Dubai |  |
| 16102 | 603 | 16-11-2020 | | 18-11-2020 | | 19:00 | | 22:00 | Australia |  |
| 16103 | 604 | 21-06-2020 | | 28-06-2020 | | 04:00 | | 07:00 | South Africa | |
| 16104 | 605 | 29-07-2020 | | 04-08-2020 | | 06:00 | | 10:00 | France |  |
| 16105 | 606 | 12-08-2021 | | 05-09-2020 | | 05:00 | | 07:30 | Germany |  |

**Functional Dependencies:**

**Ticket:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ticket\_ID -> ticket\_num |  |  |  |  |
| ticket\_num ->date\_avail,date\_flight,time\_depart,time\_land,destination | | | | |

**NORMALISATION :**

**Ticket:-**

Here,

ticket\_ID is Primary Key

ticket\_ID is candidate key

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is not Normalised to 3rd Normal Form as condition/Functional Dependency= Candidate Kay->NonPrime is there .

So we will Decompose the table into two tables.

Decomposed Table:

Table 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ticket\_ID | ticket\_num | | 16100 | 601 | | 16101 | 602 | | 16102 | 603 | | 16103 | 604 | | 16104 | 605 | | 16105 | 606 | |
|  |
|  |
|  |
|  |

Table 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | ticket\_num | date\_avail | date\_flight | time\_depart | time\_land | destination | | | 601 | 30-10-2020 | 31-10-2020 | 15:30 | 19:10 | America |  | | 602 | 16-02-2021 | 07-02-2021 | 21:00 | 23:45 | Dubai |  | | 603 | 16-11-2020 | 18-11-2020 | 19:00 | 22:00 | Australia |  | | 604 | 21-06-2020 | 28-06-2020 | 04:00 | 07:00 | South Africa | | | 605 | 29-07-2020 | 04-08-2020 | 06:00 | 10:00 | France |  | | 606 | 12-08-2021 | 05-09-2020 | 05:00 | 07:30 | Germany |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ticket TABLE | | |  | |  | |
| ticket\_ID | ticket\_num | date\_avail | | date\_flight | | time\_depart | | time\_land | destination | |
| 16100 | 601 | 30-10-2020 | | 31-10-2020 | | 15:30 | | 19:10 | America |  |
| 16101 | 602 | 16-02-2021 | | 07-02-2021 | | 21:00 | | 23:45 | Dubai |  |
| 16102 | 603 | 16-11-2020 | | 18-11-2020 | | 19:00 | | 22:00 | Australia |  |
| 16103 | 604 | 21-06-2020 | | 28-06-2020 | | 04:00 | | 07:00 | South Africa | |
| 16104 | 605 | 29-07-2020 | | 04-08-2020 | | 06:00 | | 10:00 | France |  |
| 16105 | 606 | 12-08-2021 | | 05-09-2020 | | 05:00 | | 07:30 | Germany |  |

**Functional Dependencies:**

**Ticket:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ticket\_ID -> ticket\_num |  |  |  |  |
| ticket\_num ->date\_avail,date\_flight,time\_depart,time\_land,destination | | | | |

**NORMALISATION :**

**Ticket:-**

Here,

ticket\_ID is Primary Key

ticket\_ID is candidate key

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is not Normalised to 3rd Normal Form as condition/Functional Dependency= Candidate Kay->NonPrime is there .

So we will Decompose the table into two tables.

Decomposed Table:

Table 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ticket\_ID | ticket\_num | | 16100 | 601 | | 16101 | 602 | | 16102 | 603 | | 16103 | 604 | | 16104 | 605 | | 16105 | 606 | |
|  |
|  |
|  |
|  |

Table 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | ticket\_num | date\_avail | date\_flight | time\_depart | time\_land | destination | | | 601 | 30-10-2020 | 31-10-2020 | 15:30 | 19:10 | America |  | | 602 | 16-02-2021 | 07-02-2021 | 21:00 | 23:45 | Dubai |  | | 603 | 16-11-2020 | 18-11-2020 | 19:00 | 22:00 | Australia |  | | 604 | 21-06-2020 | 28-06-2020 | 04:00 | 07:00 | South Africa | | | 605 | 29-07-2020 | 04-08-2020 | 06:00 | 10:00 | France |  | | 606 | 12-08-2021 | 05-09-2020 | 05:00 | 07:30 | Germany |  | |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Transaction TABLE | | |  | |  | | |
| trans\_id | trans\_name | cust\_ID | | trans\_date | | sched\_ID | transty\_ID | |
| 17100 | Paytm | 11100 | | 05-02-2020 | | 15100 | 18100 | |
| 17101 | Gpay | 11101 | | 15-02-2020 | | 15101 | 18101 | |
| 17102 | Bhim-UPI | 11102 | | 03-03-2020 | | 15102 | 18102 | |
| 17103 | RazourPay | 11103 | | 22-03-2020 | | 15103 | 18103 | |
| 17104 | Paytm | 11104 | | 02-07-2020 | | 15104 | 18104 | |
| 17105 | Netbanking | 11105 | | 09-12-2020 | | 15105 | 18105 | |

**Functional Dependencies:**

**Transaction:-**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | trans\_ID -> cust\_ID |  |  | | transty\_ID -> trans\_name,trans\_date,trans\_ID | | | | sched\_ID -> trans\_ID |  |  | |  |  |  |

**NORMALISATION :**

**Transaction:-**

Here,

cust\_ID is Primary Key

trans\_ID is Primary key

sched\_ID is foreign key

transty\_ID is foreign key

transty\_ID is Candidate Key

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is not Normalised to 3rd Normal Form as condition/Functional Dependency= Non Prime->Prime is there.

So we will Decompose the table into two tables.

Decomposed Table:

Table 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | trans\_id | cust\_ID | | 17100 | 11100 | | 17101 | 11101 | | 17102 | 11102 | | 17103 | 11103 | | 17104 | 11104 | | 17105 | 11105 | |
|  |
|  |
|  |
|  |

Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| transty\_ID | trans\_id | trans\_name | trans\_date |
| 18100 | 17100 | Paytm | 05-02-2020 |
| 18101 | 17101 | Gpay | 15-02-2020 |
| 18102 | 17102 | Bhim-UPI | 03-03-2020 |
| 18103 | 17103 | RazourPay | 22-03-2020 |
| 18104 | 17104 | Paytm | 02-07-2020 |
| 18105 | 17105 | Netbanking | 09-12-2020 |

Table 3

|  |  |
| --- | --- |
| sched\_ID | trans\_id |
| 15100 | 17100 |
| 15101 | 17101 |
| 15102 | 17102 |
| 15103 | 17103 |
| 15104 | 17104 |
| 15105 | 17105 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Transaction Type TABLE | | |  | |  | |
| transty\_ID | ticket\_ID | trans\_name | | sched\_ID | |
| 18100 | 16100 | Paytm | | 15100 | |
| 18101 | 16101 | Gpay | | 15101 | |
| 18102 | 16102 | Bhim-UPI | | 15102 | |
| 18103 | 16103 | RazourPay | | 15103 | |
| 18104 | 16104 | Paytm | | 15104 | |
| 18105 | 16105 | Netbanking | | 15105 | |

**Functional Dependencies:**

**Transaction Type:-**

ticket\_ID->transty\_Id

transty\_ID->trans\_name

sched\_ID->ticket\_ID

**NORMALISATION :**

**Transaction Type:-**

Here,

transty\_ID is Primary Key

sched\_ID is foreign key

ticket\_ID is foreign key

1NF -> This table is already Normalised to 1st Normal Form as it has no multivalued attributes/Atomicity.

2NF -> This table is already Normalised to 2nd Normal Form as it is in 1NF and it has no partial Dependency.

3NF -> This table is not Normalised to 3rd Normal Form as Transitive Dependency is there.

So we will Decompose the table into three tables.

Decomposed Table:

Table 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | transty\_ID | ticket\_ID | | 18100 | 16100 | | 18101 | 16101 | | 18102 | 16102 | | 18103 | 16103 | | 18104 | 16104 | | 18105 | 16105 | |
|  |
|  |
|  |
|  |

Table 2

|  |  |
| --- | --- |
| transty\_ID | trans\_name |
| 18100 | Paytm |
| 18101 | Gpay |
| 18102 | Bhim-UPI |
| 18103 | RazourPay |
| 18104 | Paytm |
| 18105 | Netbanking |

Table 3

|  |  |
| --- | --- |
| sched\_ID | ticket\_ID |
| 15100 | 16100 |
| 15101 | 16101 |
| 15102 | 16102 |
| 15103 | 16103 |
| 15104 | 16104 |
| 15105 | 16105 |

**QUERIES**

1). **CREATING TABLE RESERVATION**

create table Reservation(res\_ID int PRIMARY KEY, cust\_ID1 int,FOREIGN KEY(cust\_ID1) references Customer(cust\_ID),admin\_ID1 int, FOREIGN KEY(admin\_ID1) references admin(admin\_ID),ticket\_ID1 int,FOREIGN KEY(ticket\_ID1) references Ticket(ticket\_ID),date\_reserve date, date\_accom date,reservation\_price int);

INSERT ALL

INTO Reservation values(13100,11100,10100,16100,'05FEB20','20MAR20',20000)

INTO Reservation values(13101,11101,10101,16101,'15FEB20','21MAR20',10000)

INTO Reservation values(13102,11102,10102,16102,'03MAR20','22MAR20',50000)

INTO Reservation values(13103,11103,10103,16103,'22MAR20','23MAR20',80000)

INTO Reservation values(13104,11104,10104,16104,'02JUL20','24MAR20',30000)

INTO Reservation values(13105,11105,10105,16105,'09DEC20','25MAR20',15000)

select \* from dual;

**CREATING TABLE REPORTS**

create table Reports(report\_ID int PRIMARY KEY,trans\_ID1 int, FOREIGN KEY(trans\_ID1) references Transaction(trans\_ID),res\_ID1 int, FOREIGN KEY(res\_ID1) references Reservation(res\_ID),report\_date date);

insert all

into Reports values(12100,17100,13100,'21OCT20')

into Reports values(12101,17101,13101,'01NOV20')

into Reports values(12102,17102,13102,'02NOV20')

into Reports values(12103,17103,13103,'09NOV20')

into Reports values(12104,17104,13104,'12DEC20')

into Reports values(12105,17105,13105,'03JAN20')

select \* from dual;

PERMORMING INNER JOIN QUERY

Extracting all data where reservation Price is greater than 20000

Select \* from reservation JOIN reports on reservation.res\_ID = reports.res\_ID where reservation\_price > 20000;

2). **CREATING TABLE ADMIN**

create table admin(

admin\_ID int PRIMARY KEY,

fname varchar(20),

lname varchar(20),

gender varchar(20),

age int,

contact\_email varchar(20),

admin\_pass varchar(20)

);

INSERT ALL

INTO admin VALUES(10100,'Abdul','Aggarwal','M',18,'abdul@gmail.com','a121010')

INTO admin VALUES(10101,'Ishani','Gupta','F',20,'ishani@gmail.com','a121010')

INTO admin VALUES(10102,'Ajay','Fukta','M',25,'ajay@gmail.com','a121010')

INTO admin VALUES(10103,'Harshita','Goel','F',29,'harshita@gmail.com','a121010')

INTO admin VALUES(10104,'Gaurav','Garg','M',32,'gaurav@gmail.com','a121010')

INTO admin VALUES(10105,'Garry','Bulgurjot','M',22,'garry@gmail.com','a121010')

select \* from dual;

**CREATING TABLE CUSTOMER**create table Customer(

cust\_ID int PRIMARY KEY,

fname varchar(20),

lname varchar(20),

gender varchar(20),

age int,

contact\_add varchar(20),

cust\_email varchar(20),

cust\_pass int

);

INSERT ALL

into Customer values(11100,'Rahul','Sharma','M',25,'New Delhi','rahul@gmail.com',141010)

into Customer values(11101,'Abhay','Deol','M',22,'Mumbai','abhay@gmail.com',141011)

into Customer values(11102,'Priya','Sachdeva','F',26,'Rajasthan','priya@gmail.com',141012)

into Customer values(11103,'Raghav','Khurrana','M',22,'Banglore','raghav@gmail.com',141013)

into Customer values(11104,'Shivani','Verma','F',20,'Kerela','shivani@gmail.com',141014)

into Customer values(11105,'Diya','Sharma','F',28,'Punjab','diya@gmail.com',141015)

select \* from dual;

PERMORMING LEFT OUTER JOIN QUERY

Extracting all data where Admin age is greater than 20

Select \* from admin left outer join customer on (admin.age=customer.age) where age > 20;

3). **CREATING TABLE TRANSACTION**

create table Transaction(

trans\_ID int PRIMARY KEY,

trans\_name varchar(20),

cust\_ID2 int,

FOREIGN KEY(cust\_ID2) references Customer(cust\_ID),

trans\_date date,

sched\_ID int,

FOREIGN KEY(sched\_ID2) references Schedule(sched\_ID),

transty\_ID1 int,

FOREIGN KEY(transty\_ID1) references TransactionType(transty\_ID)

);

INSERT ALL

into Transaction values(17100,'Paytm',11100,'05FEB20',15100,18100)

into Transaction values(17101,'Gpay',11101,'15FEB20',15101,18101)

into Transaction values(17102,'Bhim-UPI',11102,'03MAR20',15102,18102)

into Transaction values(17103,'RazourPay',11103,'22MAR20',15103,18103)

into Transaction values(17104,'Paytm',11104,'02SEP20',15104,18104)

into Transaction values(17105,'Netbanking',11105,'09DEC20',15105,18105)

select \* from dual;

**CREATING TABLE TRANSACTION TYPE**create table TransactionType(

transty\_ID int PRIMARY KEY,

ticket\_ID2 int,

FOREIGN KEY(ticket\_ID2) references Ticket(ticket\_ID),

trans\_name varchar(20),

sched\_ID1 int

FOREIGN KEY(sched\_ID1) references Schedule(sched\_ID),

);

INSERT ALL

into TransactionType values(18100,16100,'Paytm',15100)

into TransactionType values(18101,16101,'Gpay',15101)

into TransactionType values(18102,16102,'Bhim-UPI',15102)

into TransactionType values(18103,16103,'Razourpay',15103)

into TransactionType values(18104,16104,'Paytm',15104)

into TransactionType values(18105,16105,'Netbanking',15105)

select \* from dual;

PERMORMING RIGHT OUTER JOIN QUERY

Extracting all data where Transaction Type ID is equal to 18104

Select \* from transaction right outer join transaction\_type on (transaction.transty\_ID=transaction\_type.transty\_ID) where transty\_ID = 18104;

4). **CREATING TABLE TRANSACTION**

create table Transaction(

trans\_ID int PRIMARY KEY,

trans\_name varchar(20),

cust\_ID2 int,

FOREIGN KEY(cust\_ID2) references Customer(cust\_ID),

trans\_date date,

sched\_ID int,

FOREIGN KEY(sched\_ID2) references Schedule(sched\_ID),

transty\_ID1 int,

FOREIGN KEY(transty\_ID1) references TransactionType(transty\_ID)

);

INSERT ALL

into Transaction values(17100,'Paytm',11100,'05FEB20',15100,18100)

into Transaction values(17101,'Gpay',11101,'15FEB20',15101,18101)

into Transaction values(17102,'Bhim-UPI',11102,'03MAR20',15102,18102)

into Transaction values(17103,'RazourPay',11103,'22MAR20',15103,18103)

into Transaction values(17104,'Paytm',11104,'02SEP20',15104,18104)

into Transaction values(17105,'Netbanking',11105,'09DEC20',15105,18105)

select \* from dual;

**CREATING TABLE TRANSACTION TYPE**create table TransactionType(

transty\_ID int PRIMARY KEY,

ticket\_ID2 int,

FOREIGN KEY(ticket\_ID2) references Ticket(ticket\_ID),

trans\_name varchar(20),

sched\_ID1 int

FOREIGN KEY(sched\_ID1) references Schedule(sched\_ID),

);

INSERT ALL

into TransactionType values(18100,16100,'Paytm',15100)

into TransactionType values(18101,16101,'Gpay',15101)

into TransactionType values(18102,16102,'Bhim-UPI',15102)

into TransactionType values(18103,16103,'Razourpay',15103)

into TransactionType values(18104,16104,'Paytm',15104)

into TransactionType values(18105,16105,'Netbanking',15105)

select \* from dual;

PERMORMING FULL OUTER JOIN QUERY

Extracting all data where Transaction Type ID is equal to 18104

Select \* from transaction full outer join transaction\_type on (transaction.transty\_ID=transaction\_type.transty\_ID);

5). **CREATING TABLE TRANSACTION**

create table Transaction(

trans\_ID int PRIMARY KEY,

trans\_name varchar(20),

cust\_ID2 int,

FOREIGN KEY(cust\_ID2) references Customer(cust\_ID),

trans\_date date,

sched\_ID int,

FOREIGN KEY(sched\_ID2) references Schedule(sched\_ID),

transty\_ID1 int,

FOREIGN KEY(transty\_ID1) references TransactionType(transty\_ID)

);

INSERT ALL

into Transaction values(17100,'Paytm',11100,'05FEB20',15100,18100)

into Transaction values(17101,'Gpay',11101,'15FEB20',15101,18101)

into Transaction values(17102,'Bhim-UPI',11102,'03MAR20',15102,18102)

into Transaction values(17103,'RazourPay',11103,'22MAR20',15103,18103)

into Transaction values(17104,'Paytm',11104,'02SEP20',15104,18104)

into Transaction values(17105,'Netbanking',11105,'09DEC20',15105,18105)

select \* from dual;

**CREATING TABLE TRANSACTION TYPE**create table TransactionType(

transty\_ID int PRIMARY KEY,

ticket\_ID2 int,

FOREIGN KEY(ticket\_ID2) references Ticket(ticket\_ID),

trans\_name varchar(20),

sched\_ID1 int

FOREIGN KEY(sched\_ID1) references Schedule(sched\_ID),

);

INSERT ALL

into TransactionType values(18100,16100,'Paytm',15100)

into TransactionType values(18101,16101,'Gpay',15101)

into TransactionType values(18102,16102,'Bhim-UPI',15102)

into TransactionType values(18103,16103,'Razourpay',15103)

into TransactionType values(18104,16104,'Paytm',15104)

into TransactionType values(18105,16105,'Netbanking',15105)

select \* from dual;

PERMORMING RIGHT OUTER JOIN QUERY

Extracting all data where Transaction Type ID is equal to 18104

Select \* from transaction right outer join transaction\_type on (transaction.transty\_ID=transaction\_type.) where transty\_ID = 18104;

6). **CREATING TABLE TRANSACTION**

create table Transaction(

trans\_ID int PRIMARY KEY,

trans\_name varchar(20),

cust\_ID2 int,

FOREIGN KEY(cust\_ID2) references Customer(cust\_ID),

trans\_date date,

sched\_ID int,

FOREIGN KEY(sched\_ID2) references Schedule(sched\_ID),

transty\_ID1 int,

FOREIGN KEY(transty\_ID1) references TransactionType(transty\_ID)

);

INSERT ALL

into Transaction values(17100,'Paytm',11100,'05FEB20',15100,18100)

into Transaction values(17101,'Gpay',11101,'15FEB20',15101,18101)

into Transaction values(17102,'Bhim-UPI',11102,'03MAR20',15102,18102)

into Transaction values(17103,'RazourPay',11103,'22MAR20',15103,18103)

into Transaction values(17104,'Paytm',11104,'02SEP20',15104,18104)

into Transaction values(17105,'Netbanking',11105,'09DEC20',15105,18105)

select \* from dual;

**CREATING TABLE Schedule**create table Schedule(

sched\_ID int PRIMARY KEY,

date\_flight date,

time\_depart timestamp,

time\_land timestamp

);

INSERT ALL

into Schedule values(15100,'21OCT20','15:30:00','19:10:30')

into Schedule values(15101,'01NOV20','21:00:00','23:45:50')

into Schedule values(15102,'02NOV20','19:00:05','22:00:30')

into Schedule values(15103,'09DEC20','16:00:30','19:00:22')

into Schedule values(15104,'12DEC20','18:00:22','10:00:00')

into Schedule values(15105,'03JAN21','17:00:00','19:30:25')

select \* from dual;

PERMORMING INTERSECTION QUERY

Extracting all data where Transaction Type ID is equal to 18104

Select \* from transaction INTERSECT select \* from schedule;

7). **CREATING TABLE ADMIN**

create table admin(

admin\_ID int PRIMARY KEY,

fname varchar(20),

lname varchar(20),

gender varchar(20),

age int,

contact\_email varchar(20),

admin\_pass varchar(20)

);

INSERT ALL

INTO admin VALUES(10100,'Abdul','Aggarwal','M',18,'abdul@gmail.com','a121010')

INTO admin VALUES(10101,'Ishani','Gupta','F',20,'ishani@gmail.com','a121010')

INTO admin VALUES(10102,'Ajay','Fukta','M',25,'ajay@gmail.com','a121010')

INTO admin VALUES(10103,'Harshita','Goel','F',29,'harshita@gmail.com','a121010')

INTO admin VALUES(10104,'Gaurav','Garg','M',32,'gaurav@gmail.com','a121010')

INTO admin VALUES(10105,'Garry','Bulgurjot','M',22,'garry@gmail.com','a121010')

select \* from dual;

**CREATING TABLE CUSTOMER**create table Customer(

cust\_ID int PRIMARY KEY,

fname varchar(20),

lname varchar(20),

gender varchar(20),

age int,

contact\_add varchar(20),

cust\_email varchar(20),

cust\_pass int

);

INSERT ALL

into Customer values(11100,'Rahul','Sharma','M',25,'New Delhi','rahul@gmail.com',141010)

into Customer values(11101,'Abhay','Deol','M',22,'Mumbai','abhay@gmail.com',141011)

into Customer values(11102,'Priya','Sachdeva','F',26,'Rajasthan','priya@gmail.com',141012)

into Customer values(11103,'Raghav','Khurrana','M',22,'Banglore','raghav@gmail.com',141013)

into Customer values(11104,'Shivani','Verma','F',20,'Kerela','shivani@gmail.com',141014)

into Customer values(11105,'Diya','Sharma','F',28,'Punjab','diya@gmail.com',141015)

select \* from dual;

PERMORMING SUBQUERY QUERY

Extracting all data from Admin Table where customer age is > admin age

Select admin\_ID from admin where age > ANY(select age from customer where age>20);

8). **CREATING TABLE Schedule**

create table Schedule(

sched\_ID int PRIMARY KEY,

date\_flight date,

time\_depart timestamp,

time\_land timestamp

);

INSERT ALL

into Schedule values(15100,'21OCT20','15:30:00','19:10:30')

into Schedule values(15101,'01NOV20','21:00:00','23:45:50')

into Schedule values(15102,'02NOV20','19:00:05','22:00:30')

into Schedule values(15103,'09DEC20','16:00:30','19:00:22')

into Schedule values(15104,'12DEC20','18:00:22','10:00:00')

into Schedule values(15105,'03JAN21','17:00:00','19:30:25')

select \* from dual;

PERMORMING PL/SQL QUERY

DECLARE

Sched\_ID schedule.sched\_ID%type;

Date\_flight schedule.date\_flight%type;

Time\_depart schedule.time\_depart%type;

Time\_land schedule.time\_land%type;

BEGIN

INSERT ALL

into Schedule values(15100,'21OCT20','15:30:00','19:10:30')

into Schedule values(15101,'01NOV20','21:00:00','23:45:50')

into Schedule values(15102,'02NOV20','19:00:05','22:00:30')

into Schedule values(15103,'09DEC20','16:00:30','19:00:22')

into Schedule values(15104,'12DEC20','18:00:22','10:00:00')

into Schedule values(15105,'03JAN21','17:00:00','19:30:25')

select \* from dual;

UPDATE schedule SET time\_depart=’20:30:09’ where time\_land=’19:30:25’;

END;

/

8). **CREATING TABLE Customer**

create table Customer(

cust\_ID int PRIMARY KEY,

fname varchar(20),

lname varchar(20),

gender varchar(20),

age int,

contact\_add varchar(20),

cust\_email varchar(20),

cust\_pass int

);

INSERT ALL

into Customer values(11100,'Rahul','Sharma','M',25,'New Delhi','rahul@gmail.com',141010)

into Customer values(11101,'Abhay','Deol','M',22,'Mumbai','abhay@gmail.com',141011)

into Customer values(11102,'Priya','Sachdeva','F',26,'Rajasthan','priya@gmail.com',141012)

into Customer values(11103,'Raghav','Khurrana','M',22,'Banglore','raghav@gmail.com',141013)

into Customer values(11104,'Shivani','Verma','F',20,'Kerela','shivani@gmail.com',141014)

into Customer values(11105,'Diya','Sharma','F',28,'Punjab','diya@gmail.com',141015)

select \* from dual;

PERMORMING TRIGGER ON INSERT STATEMENT

Create or replace trigger cust\_trigger before insert on customer for each row

DECLARE

Cust\_except EXCEPTION;

Pragma EXCEPTION \_init(cust\_excep,-2001);

BEGIN IF :NEW.age < 0 THEN

Raise \_Application\_ERROR(-2001,’Age Cannot be negative’);

ELSE

DBMS\_output.put\_line(‘Data Inserted’);

ENDIF

END;

/