0



A DATABASE DESIGN FOR METATRAVEL

by Dataverse:

RAZAN AL HAJJAR

CYNTHIA FRANCIS

AHMAD HUSSEIN

ALI SALMAN

Submitted to:

DR. RAMZI HARATY in partial fulfillment of the requirements for the course “CSC375: Database Management Systems”

Monday December 5, 2022

1. **Table of Contents:**
2. Copyright Notice…………………………………………………………………………………………………..…..1
3. Introduction…………………………………………………………………………………………………………..….….2
4. System Description and Constraints…………………………………………………..….……..…3
5. ER Diagram Symbols………………………………………………………………………………….……...….4
6. ER Diagram for Metatravel………………………………………………………………………………....6
7. Entity Types…………………………………………………………………………………………………………...……7
8. Relationships………………………………………………………………………………………….…………..……...18
9. ER to Relational Mapping Algorithms………………………………….………………..….….26
10. Mapping Result…………………………………..…………………………….………………………….…………..37
11. SQL Queries and Oracle Server……………………………….………………….……………….…..40
12. Table Descriptions…………………………………..…………….………………………….……………….…...47
13. Inserting Data…………………………………..…………………………….………………………….………………54
14. Final Table State………..………………………………………….……………….………………………….…… 71
15. Queries: …………………………………..…………………………….………………………….……………….………..84
16. Normalization Up to BCNF…….………………………….………………………….……….…….…..111
17. References…………………………………..………….………………….………………………….……………….….121
18. **Copyright Notice**

Copyright © 2022-2023. Dataverse. All rights reserved. This work is protected by Copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission(s) to use material from this work, please submit a written request to one of the authors of this work: razan.alhajjar@lau.edu, cynthia.francis@lau.edu, [ahmad.hussein03@lau.edu](mailto:ahmad.hussein03@lau.edu), alielhadi.salman@lau.edu.

1. **Introduction**

As the world is evolving, technology has accelerated its pace and has integrated itself in humans' daily lives. To keep up with this pace and stay applicable, several companies have begun mapping out their futuristic plans. For instance, Meta – formerly known as Facebook – has announced its plan of creating a metaverse.

While the metaverse might seem like a relatively new concept, the idea of virtual reality is not. In fact, the term “metaverse” dates back to 1982 in Neil Stevenson's science fiction novel “Snow Crash.” The characters from the novel escape from dystopian Los Angeles, plagued by hyperinflation, inequality, and illness, and immerse themselves into a virtual world called the “metaverse” as an escape from their crumbling society (Zenou, 2022).

Inspired by the novel, Meta's CEO, Mark Zuckerberg, announced his vision of people inhabiting digital environments during the company's rebranding event in October 2021 (Kelly, 2021). During that event, Meta shared its vision of building the metaverse, where users can (virtually) socialize, work, and even travel.

With the intention of bringing this vision to life, the Dataverse team was assigned to create a database for the company's metaverse travel agency, called “Metatravel.” The agency's main role is to arrange users' travel inside the metaverse.

*Ladies and gentlemen, on behalf of the co-captains, welcome onboard. Please make sure that your seat is comfortable, and your Meta-helmet is correctly fastened. Please turn off all personal electronic devices, including laptops and cell phones. The travel through the database design is about to begin. Hope you enjoy your trip with us.*

1. **System Description and Constraints**

Metatravel is a virtual travel agency in the metaverse. The agency's mission statement is, “to inspire travelers – in air, space, on water, and land – all through their screens.” Metatravel's role is to assist its customers with their travel arrangements and concerns.

The agency is spread across several branches in different locations. It is composed of various departments, each attending to our customers – whether it be on or behind the scenes. Well-trained travel agents are in direct contact with customers. Each user is assigned an agent to help them choose the destination, set a budget, purchase travel tickets, and book accommodations. Other employees, such as financial managers and accountants, work behind the scenes to ensure that all of the customer's needs are met. Employees are remunerated for their work in cryptocurrency, namely Bitcoin (BTC).

With Metatravel, users have the freedom to choose the mode of transportation that best suits them – flights, trains, cruises, or spaceships. Furthermore, through the dealerships established with partners, users are provided with tickets and/or accommodations at reduced costs. Some of the partners include Meta-Hotel, Meta-Airways, Meta-Railways, Meta-Cruises, and Meta-Space.

First, users contact a branch of the travel agency, providing it with their personal information - name, date of birth, passport ID, address, and emergency contact. In the interim, an employee drafts a contract for the user to sign with an expiration date.

Second, upon agreeing to the contract legalities, the user must electronically sign it. Some contracts may be refundable. Post-signing the binding contract, the customer settles fees and gains loyalty points for choosing Metatravel. According to purchase history and those loyalty points, discounts are applied with a certain formula. With every trip, the user gains five points; 35 points lead to 15% discount on the total cost; 70 points lead to 30% discount; 100 points lead to 50% discount. After that, the points are reset.

After purchasing the ticket, the user, with the agent's guidance, books the hotel room of their preference. In each hotel, miscellaneous meal plans are offered. For reference, star-rating is provided for every hotel as additional guidance for the user.

1. **ER Diagram Symbols**

To ease your understanding of the ER diagram, ER diagram symbol annotation is inserted below.

|  |  |
| --- | --- |
| **Name** | **Symbol** |
| Entity Type |  |
| Weak Entity Type |  |
| Attribute |  |
| Key Attribute |  |
| Multivalued Key |  |
| Partial Key | Partial Key |
| Derived Attribute |  |
| Composite Attribute |  |
| Relationship |  |
| Identifying Relationship |  |
| Total Participation |  |
| Partial Participation |  |

1. **ER Diagram for Metatravel[[1]](#footnote-1)**



1. **Entity Types**
2. **Customer:**



The **CUSTOMER** is the metaverse user who plans on traveling. He or she **CONTACTS** a certain **BRANCH** of Meta-travel. The **CUSTOMER** **SIGNS** the **CONTRACT**, **BOOKS** a **TICKET**, **PAYS** the **BILL**, and **STAYS\_AT** a **HOTEL**.

The **CUSTOMER**'s key attribute is his or her Passport\_id. All of the **CUSTOMER**'s attributes are listed below:

* Passport\_id: Consists of 9 digits.
* Name: Specifies the customer's name as a composite attribute consisting of the First\_Name, Middle\_Name, and Family\_Name.
* Chat\_id: Consists of the 8-digit number the customer uses to communicate
* Date\_Of\_Birth: Mentions the user's date of birth in the format: MM/DD/YYYY
* Age: Is a derived attribute from Date\_Of\_Birth.
* Address: Stores the user's address as a composite attribute consisting of the Street\_Number, City, State, Country, and Planet.
* Gender: Specifies the user's gender.
* Purchase\_Tickets\_id: Stores the IDs of the customer's previously purchased tickets as a multivalued attribute.
* Points\_balance: Shows the customer's loyalty points balance.

1. **Employee**



Each **EMPLOYEE** **MANAGES** a **CUSTOMER**. The **EMPLOYEE** **DRAFTS** a **CONTRACT**, **RECOMMENDS** a **HOTEL**, and **WORKS\_FOR** a **DEPARTMENT**. An **EMPLOYEE SUPERVISES** another **EMPLOYEE** as a *recursive or self-referencing relationship type.*

The **EMPLOYEE**'s key attribute is his or her Essn. All of the **EMPLOYEE**'s attributes are listed below:

* Essn: Consists of the employee's social security number of 9 digits.
* Name: Specifies the employee's name as a composite attribute consisting of the First\_Name, Middle\_Name, and Family\_Name.
* Chat\_id: Consists of the 8-digit number the employee uses to communicate
* Date\_Of\_Birth: Mentions the employee's date of birth in the format: MM/DD/YYYY.
* Age: Is a derived attribute from Date\_Of\_Birth.
* Address: Stores the employee's address as a composite attribute consisting of the Street\_Number, City, State, Country, and Planet.
* Gender: Specifies the employee's gender.
* Salary: Includes the employee's monthly salary in Bitcoin (BTC).
* Position: Identifies the employee's work position in the agency.

1. **Ticket**



After the **PARTNER** **PROVIDES** the **TICKET**, the **CUSTOMER BOOKS** the **TICKET** as a form of verification for travelling. A **TICKET CONNECTS TO** another **TICKET** as a *recursive or self-referencing relationship type.* Each relationship instance relates two distinct **TICKETs** in the case of connecting tickets. Sometimes, the **CUSTOMER** might not be able to reach destination in a single trip. This gap is filled by connecting **TICKETs**.

The **TICKET**'s key attribute is the Ticket\_number. All of the **TICKET**'s attributes are listed below:

* Ticket\_number: Consists of the unique 7-digit number corresponding to each ticket purchased.
* Mode\_Of\_Transportation: Includes the mode of transportation chosen.
* Transport\_Company: Mentions the name of the company providing the transportation service. Every transport company provides only one mode of transportation.
* Baggage\_Weight: Specifies the maximum permitted weight of the baggage.
* Seat\_Number: Stores the seat number given to the customer.
* Departure: Is a composite attribute composed of the departure date, time and location [as a composite attribute of the City, State, Country, Planet].
* Destination: Is a composite attribute composed of the arrival date, time and location [as a composite attribute of the City, State, Country, Planet].
* Price: States the ticket's price in Bitcoin (BTC).
* Ticket\_Type: Describes whether the ticket is one-way or round-trip.
* Is\_Booked: Specifies whether the ticket is the booked status or not.

1. **Hotel**



The **HOTEL** is the accommodation the **EMPLOYEE** **RECOMMENDS**,and the **CUSTOMER** chooses to **STAY\_AT.**

The **HOTEL**'s key attribute is the Hotel\_id. All of the **HOTEL**'s attributes are listed below:

* Hotel\_id: Consists of the unique id for each hotel.
* Hotel\_Name: Mentions the hotel's name.
* Address: Stores the hotel's address as a composite attribute consisting of the Street\_Number, City, State, Country, and Planet.
* Star\_Rating: Specifies the number of star rating given to the hotel.
* Number\_Of\_Meals: Includes the number of meals provided per day.
* Price: States the hotel's price per night in Bitcoin (BTC).
* Is\_Partner: Mentions whether the hotel is a partner of Metatravel agency or not.

1. **Room**



The entity **ROOM** is a weak entity type that is identification-dependent on the identifying entity **HOTEL**. The partial key of **ROOM** is the Room\_Number. All of the **ROOM**'s attributes are listed below:

* Room\_Number: Specifies the room number in the hotel booked.
* Capacity: Includes the maximum number of people permitted per room.
* Room\_Type: Describes whether the room is single-bed, double-bed, triple-bed, king-sized bed, queen-sized bed, or a suite.

1. **Companion**



The entity **COMPANION** is a weak entity type that is identification-dependent on the identifying entity **CUSTOMER**. The **COMPANION** **ACCOMPANIES** the **CUSTOMER**. The partial key of **COMPANION** is the Name. All of the **COMPANION**'s attributes are listed below:

* Name: Specifies the companion's name as a composite attribute consisting of the First\_Name, Middle\_Name, and Family\_Name.
* Chat\_id: Consists of the 8-digit number the companion uses to communicate
* Date\_Of\_Birth: Mentions the companion's date of birth in the format: MM/DD/YYYY.
* Age: Is a derived attribute from Date\_Of\_Birth.
* Gender: Specifies the companion's gender.
* Relationship: Includes the relationship between the companion and customer.

1. **Contract**



The **EMPLOYEE** **DRAFTS** the **CONTRACT** and the **CUSTOMER** **SIGNS** it, thereby formalizing the relationship between the agency and the customer, outlining the various legal obligations each party owes to the other, and finalizing payments.

The **CONTRACT**'s key attribute is the Contract\_id. All of the **CONTRACT**'s attributes are listed below:

* Contract\_id: Consists of the unique id for each contract.
* Is\_Refundable: Mentions whether refund for the payment is applicable.
* Is\_Signed: Indicates whether the customer signed the contract.
* Expiry\_Date: Specifies the date wherein the contract ends.
* Travel\_Reason: Specifies the reason the customer wants to visit a specific location.

1. **Bill**



The **CUSTOMER** **PAYS** the **BILL**. The **BILL**'s key attribute is the Transaction\_id. All of the **CONTRACT**'s attributes are listed below:

* Transaction\_ID: Consists of the unique id for each transaction.
* Payment\_Date: Indicates the date in which the final payment is made, and no further exchanges will occur.
* Method\_Of\_payment: Specifies the payment method.
* Amount: Is a derived attribute from TICKET and HOTEL price.
* Points\_Discount: Determines the amount of money deducted from the payment. It is a derived attribute from the CUSTOMER's Points\_balance.

1. **Department**



The **EMPLOYEE WORKS** **FOR** a specific **DEPARTMENT**, and a **BRANCH HAS\_A** a **DEPARTMENT**. **DEPARTMENT** a weak entity type that is identification-dependent on the identifying entity **BRANCH**. The **DEPARTMENT**'s partial key is the Department\_Name. All of the **DEPARTMENT**'s attributes are listed below:

* Department\_Name: Indicates the name of the department.
* Description: Describes the department' function.

1. **Emergency\_Contact**



The **CUSTOMER**/**EMPLOYEE HAS\_AN** **EMERGENCY\_CONTACT**. The entity **EMERGENCY\_CONTACT** is a weak entity type that is identification-dependent on the identifying entities **CUSTOMER** and **EMPLOYEE**. The partial key for **EMERGENCY\_CONTACT** is the Name: All of the **EMERGENCY\_CONTACT**'s attributes are listed below:

* Name: Specifies the emergency contact's name as a composite attribute consisting of the First\_Name, Middle\_Name, and Family\_Name.
* Chat\_id: Consists of the 8-digit number the companion uses to communicate
* Relationship: Includes the relationship between the emergency contact and customer.

1. **Branch**



The **CUSTOMER CONTACTS** a specific **BRANCH**. Moreover, each **BRANCH** **HAS\_A DEPARTMENT**. The **BRANCH**'s key attribute is the Branch\_Number. All of the **BRANCH**'s attributes are listed below:

* Branch\_Number: Consists of the 5-digit number that indicates the specific branch of the department.
* Address: Stores the branch's address as a composite attribute consisting of the Street\_Number, City, State, Country, and Planet.
* Number\_Of\_Employees: Specifies the number of employees in the branch.

1. **Partner**



Dealerships between **PARTNERs** and the agency are established in order to benefit customers. A **PARTNER PROVIDES** a **TICKET** at reduced costs. The **PARTNER**'s key attribute is the Partner\_ID. All of the **PARTNER**'s attributes are listed below:

* Partner\_ID: Specifies the partner's unique id number.
* Name: Includes the partner's name.
* Address: Stores the partner's address as a composite attribute consisting of the Street\_Number, City, State, Country, and Planet.
* Service: Describes the service provided by the partner.
* Operating\_Countries: Is a multivalued attribute which states the countries in which the partner operates.

1. **Relationships**
   1. **Manages**



The **MANAGES** relationship exists between **EMPLOYEE** and **CUSTOMER**. The relationship is *many-to-many (M:N)*; An **EMPLOYEE** may **MANAGE** many **CUSTOMERS**, and a **CUSTOMER** may be managed by many **EMPLOYEES**.

The participation constraint is partial for the **EMPLOYEE** entity type since not each **EMPLOYEE MANAGES** a **CUSTOMER**. However, the participation constraint is total for the **CUSTOMER** entity type since each **CUSTOMER** must be managed by at least one **EMPLOYEE**.

* 1. **Supervises**



The **SUPERVISES** relationship is a self-referencing relationship between an **EMPLOYEE** and another **EMPLOYEE**. The relationship is *one-to-many (1:N).* One **EMPLOYEE** **SUPERVISES** many **EMPLOYEEs**, and an **EMPLOYEE** is **SUPERVISED** by one **EMPLOYEE.**

The participation constraint is partial on both sides for the **EMPLOYEE** entity types since not every **EMPLOYEE** necessarily **SUPERVISES** another one, and not every **EMPLOYEE**, namely the CEO,is **SUPERVISED** by another one.

* 1. **Drafts**



The **DRAFTS** relationship exists between an **EMPLOYEE** and a **CONTRACT**. The relationship is *one-to-many (1:N)*; An **EMPLOYEE** may **DRAFT** several **CONTRACTs**, but a **CONTRACT** is **DRAFTED** by only one **EMPLOYEE**.

The participation constraint is partial on the **EMPLOYEE** entity side since not every **EMPLOYEE** necessarily **DRAFTS** a **CONTRACT**. On the other hand, the participation is constraint is total on the **CONTRACT** entity side since every **CONTRACT** is **DRAFTED** by an **EMPLOYEE**.

* 1. **Recommends**



**RECOMMENDS** is a ternary relationship between an **EMPLOYEE**, a **HOTEL**, and a**CUSTOMER** since the **EMPLOYEE RECOMMENDS** the **HOTEL** to the **CUSTOMER**. The relationship is *one-to-one-to-many (1:1:N)*; An **EMPLOYEE** may **RECOMMEND** several **HOTELs** to a single **CUSTOMER** according to their need, but a **HOTEL** is **RECOMMENDed** by only one **EMPLOYEE**.

The participation constraint is partial on the **EMPLOYEE** entity side since not every **EMPLOYEE** necessarily **RECOMMENDS** a **HOTEL**. Moreover, the participation is partial on the **HOTEL** entity side since not every **HOTEL** is **RECOMMENDED** by an **EMPLOYEE**. The participation constraint is also partial on the **CUSTOMER** entity side since not every **CUSTOMER** is necessarily **RECOMMENDED** a **HOTEL**

* 1. **Works\_For**



The **WORKS\_FOR** relationship exists between an **EMPLOYEE** and a **DEPARTMENT**. The relationship is *many-to-one (N:1)*; Many **EMPLOYEEs** **WORK\_FOR** one **DEPARTMENT**.

The participation constraint is total on the **EMPLOYEE** and **DEPARTMENT** entity sides since every **EMPLOYEE WORKS\_FOR** a **DEPARTMENT**, and every **DEPARTMENT** has **EMPLOYEEs** who **WORK\_FOR** it.

* 1. **Contains**



The **CONTAINS** relationship exists between a **BRANCH** and a **DEPARTMENT**. The relationship is *one-to-many (1:N)*; A **BRANCH CONTAINS** many **DEPARTMENTs**,but a **DEPARTMENT** is only **CONTAINED** in one **BRANCH**.

The participation constraint is total on the **BRANCH** and **DEPARTMENT** entity sides since every **BRANCH CONTAINS** a **DEPARTMENT**, and every **DEPARTMENT** is **CONTAINED** in a **BRANCH**.

* 1. **Books**



The **BOOKS** relationship exists between as **CUSTOMER** and a **TICKET**. The relationship is *one-to-many (1:N)*; A **CUSTOMER** may **BOOK** many **TICKETs**, but a **TICKET** may be **BOOKED** by a single **CUSTOMER**.

The participation constraint is total on the **CUSTOMER** entity side since every **CUSTOMER BOOKS** a **TICKET**. However, the participation constraint is partial on the **TICKET** entity side since not every **TICKET** is **BOOKED** by a **CUSTOMER**.

This relationship has as attribute the Booking\_Date which states the date in which the **CUSTOMER** **BOOKED** TICKET(s).

* 1. **Stays\_At**



The **STAYS\_AT** relationship exists between as **CUSTOMER** and a **HOTEL**. The relationship is *many-to-one (N:1)*; A **HOTEL** may have many **CUSTOMERs** **STAY\_AT** it, and **CUSTOMER** only **STAYS\_AT** one **HOTEL**.

The participation constraint is partial on the **CUSTOMER** and a **HOTEL** entity sides since not every **CUSTOMER** might **STAY\_AT** a **HOTEL**, and not every **HOTEL** may have a **CUSTOMER STAY\_AT** it.

This relationship has as attribute the Stay\_Duration which states the period of time the **CUSTOMER** will be **STAYING\_AT** the **HOTEL**.

* 1. **Accompanies**



The **ACCOMPANIES** relationship exists between a **CUSTOMER** and a **COMPANION**. The relationship is *one-to-many (1:N)*; A **CUSTOMER** may have several **COMPANIONs ACCOMPANYING** them, but a **COMPANION** only **ACCOMPANIES** one **CUSTOMER**.

The participation constraint is partial on the **CUSTOMER** side since not every **CUSTOMER** might be **ACCOMPANIED** by a **COMPANION**. On the other hand, the participation constraint is total on the **COMPANION** side since every **COMPANION is ACCOMPANIED by** a **CUSTOMER**.

* 1. **Signs**



The **SIGNS** relationship exists between a **CUSTOMER** and a **CONTRACT**. The relationship is *many-to-one (N:1)*; A **CUSTOMER** may **SIGN** many **CONTRACTs**, but a **CONTRACT** is only **SIGNED** by one **CUSTOMER**.

The participation constraint is total on the **CUSTOMER** entity side since every **CUSTOMER SIGNS** a **CONTRACT**. Furthermore, the participation constraint is partial on the **CONTRACT** entity side since not every **CONTRACT** **IS SIGNED** by a **CUSTOMER**.

* 1. **Pays**



The **PAYS** relationship exists between a **CUSTOMER** and a **BILL**. The relationship is *many-to-one (N:1)*; A **CUSTOMER** may **PAY** many **BILLs**, but a **BILL** is only **PAID** by one **CUSTOMER**.

The participation constraint is total on both the **CUSTOMER** and **BILL** entity sides since every **CUSTOMER PAYS** a **BILL**, and every **BILL** **IS PAID** by a **CUSTOMER** after signing the contract.

* 1. **Contacts**



The **CONTACTS** relationship exists between a **CUSTOMER** and a **BRANCH**. The relationship is *many-to-one (N:1)*; A **CUSTOMER** **CONTACTS** only one **BRANCH**, but a **BRANCH** may be **CONTACTed** by many **CUSTOMERs**.

The participation constraint is total on both the **CUSTOMER** and **BRANCH** entity sides since not every **CUSTOMER CONTACTS** a **BRANCH**, and every **BRANCH** **IS CONTACTED** by a **CUSTOMER**.

* 1. **Prepares**



The **PREPARES** relationship exists between a **HOTEL** and a **ROOM**. The relationship is *one-to-many (1:N)*; A **HOTEL** **PREPARES** several **ROOMS**, but a **ROOM** is **PREPARED** in one **HOTEL**.

The participation constraint is total on **ROOM** entity side since every **ROOM** is **PREPARED** by. The participation constraint is partial on the **HOTEL** entity side since every **HOTEL PREPARES** a **ROOM**.

* 1. **Provides**



The **PROVIDES** relationship exists between a **PARTNER** and a **TICKET**. The relationship is *one-to-many (1:N)*; A **PARTNER** **PROVIDES** several **TICKETs**, but a **TICKET** is **PROVIDEd** by one **PARTNER**.

The participation constraint is partial on the **PARTNER** entity side since not every **PARTNER**, for instance a **HOTEL**, **PROVIDES** a **TICKET**. In contrast, the participation constraint is total **TICKET** entity side since every **TICKET** is **PROVIDED** by a **PARTNER**.

* 1. **Connects\_With**



The **CONNECTS\_WITH** relationship exists between a **TICKET** and another **TICKET**. The relationship is *one-to-one (1:1)*; A **TICKET** **CONNECTS\_WITH** one **TICKET**, and a **TICKET** is **CONNECTED\_WITH** one **TICKET**.

The participation constraint is partial on both the **TICKET** entity sides since not every **TICKET CONNECTS\_WITH** another **TICKET**, and not every **TICKET** is **CONNECTED\_WITH** a **TICKET**.

* 1. **Has\_An1**



The **HAS\_AN** relationship exists between a **CUSTOMER** and an **EMERGENCY\_CONTACT**. The relationship is *one-to-many (1:N)*; A **CUSTOMER** may **HAVE** many **EMERGENCY\_CONTACT**, and an **EMERGENCY\_CONTACT** may only be used for one **CUSTOMER**.

The participation constraint is total on both the **CUSTOMER** and **EMERGENCY\_CONTACT** entity sides since every **CUSTOMER HAS\_AN** **EMERGENCY\_CONTACT** and every **EMERGENCY\_CONTACT** is used for a **CUSTOMER**.

* 1. **Has\_An2**



The **HAS\_AN** relationship exists between an **EMPLOYEE** and an **EMERGENCY\_CONTACT**. The relationship is *one-to-many (1:N)*; An **EMPLOYEE** may **HAVE** many **EMERGENCY\_CONTACT**, and an **EMERGENCY\_CONTACT** may only be used for one **EMPLOYEE**.

The participation constraint is total on both the **EMPLOYEE** and **EMERGENCY\_CONTACT** entity sides since every **EMPLOYEE HAS\_AN** **EMERGENCY\_CONTACT**, and every **EMERGENCY\_CONTACT** is used for an **EMPLOYEE.**

1. **ER to Relational Mapping Algorithms**

After designing the ER diagram and creating attributes, entities, and relationships, we will translate the ER diagram to a relational database design. A seven-step algorithm will be used to do the mapping of the ER diagram to a relational database design. The steps are as follows:

**STEP 1: Mapping of Regular Entity Types**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Passport\_id | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth |

* 1. **CUSTOMER**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Street\_Number | City | State | Country | Planet | Gender | Points\_balance |

The **CUSTOMER** entity contains simple, derived, composite and multivalued attributes. The derived attribute Age and the multivalued attribute Purchased\_Tickets\_id are not represented in this relation. This relation solely consists of simple attributes and the primary key ID which is underlined. The **CUSTOMER** entity has Name and Address as composite attributes. The simple attributes First\_Name, Last\_Name, and Middle\_Name that compose Name are included in this relation. Equivalently, Street\_Number, City, State, Country, and Planet that compose Address are included in this relation.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Essn | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth | Street\_Number | City | State |

* 1. **EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Country | Planet | Gender | Salary | Position |

The **EMPLOYEE** entity contains simple, derived, and composite attributes. The derived attribute Age is not represented in this relation. This relation solely consists of simple attributes and the primary key ID which is underlined. The **EMPLOYEE** entity has Name and Address as composite attributes. The simple attributes First\_Name, Last\_Name, and Middle\_Name that compose Name are included in this relation. Equivalently, Street\_Number, City, State, Country, and Planet that compose Address are included in this relation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ticket\_number | Mode\_of\_Transportation | Transport\_company | Baggage\_weight | Seat\_number | Destination\_City | Destination\_State |

* 1. **TICKET**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Destination\_Country | Destination\_Planet | Departure\_City | Departure\_State | Departure\_Country | Departure\_Planet | Price | Ticket\_type | Is\_booked |

|  |  |  |  |
| --- | --- | --- | --- |
| Destination\_Time | Destination\_Date | Departure\_Time | Departure\_Date |

The **TICKET** entity contains simple and composite attributes. This relation solely consists of simple attributes and the primary key ID which is underlined. The **TICKET** entity has Destination, of which Location is composite, and Departure, of which Location is composite, as composite attributes. The simple attributes Destination\_Date and Destination\_Time that compose Destination are included in this relation. Destination\_City, Destination\_State, Destination\_Country, and Destination\_Planet that compose Location are included in this relation. The simple attributes Departure\_Date and Departure\_Time that compose Departure are included in this relation. Departure\_City, Departure\_State, Departure\_Country, and Departure\_Planet that compose Location are included in this relation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Hotel\_id | Hotel\_name | Street\_Number | City | State | Country | Planet |

1. **HOTEL**

|  |  |  |  |
| --- | --- | --- | --- |
| Star\_rating | Number\_of\_Meals | Price | is\_Partner |

The **HOTEL** entity contains simple and composite attributes. This relation solely consists of simple attributes and the primary key ID which is underlined. The **HOTEL** entity has Address as composite attribute. The simple attributes Street\_Number, City, State, Country, and Planet that compose Address are included in this relation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Contract\_id | is\_Refundable | is\_Signed | Expiry\_date | Travel\_reason |

1. **CONTRACT**

The **CONTRACT** entity contains simple attributes. This relation solely consists of simple attributes and the primary key ID which is underlined.

1. **BILL**

|  |  |  |
| --- | --- | --- |
| Transaction\_id | Payment\_date | Method\_of\_Payment |

The **BILL** entity contains simple and derived attributes. The derived attributes Amount and Points\_Discount are not represented in this relation. This relation solely consists of simple attributes and the primary key ID which is underlined.

1. **BRANCH**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Branch\_Number | Street\_Number | City | State | Country | Planet |

The **BRANCH** entity contains simple and composite attributes. This relation solely consists of simple attributes and the primary key ID which is underlined. The **BRANCH** entity has Address as a composite attribute. The simple attributes Street\_Number, City, State, Country, and Planet that compose Address are represented in this relation.

1. **PARTNER**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Partner\_id | Name | City | State | Country | Planet | Service | Street\_Number |

The **PARTNER** entity contains simple, composite and multivalued attributes. The multivalued attribute Operating\_Countries are not represented in this relation. This relation solely consists of simple attributes and the primary key ID which is underlined. The **PARTNER** entity has Address as a composite attribute. The simple attributes Street\_Number, City, State, Country, and Planet that compose Address are represented in this relation.

**STEP 2: Mapping of Weak Entity Types**

* 1. **ROOM**

|  |  |  |  |
| --- | --- | --- | --- |
| Room\_Number | H\_id | Capacity | Room\_Type |

The weak entity **ROOM** contains only simple attributes. The weak entity does not have any derived, multivalued, or composite attributes. Moreover, the H\_id, the primary of the owner entity **HOTEL**, is included. H\_id and the partial key Room\_Number are combined to represent the primary key of this relation.

* 1. **COMPANION**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | P\_id | Chat\_id | Date\_of\_Birth | Gender | Relationship |

The **COMPANION** entity contains simple, derived, and composite attributes. The derived attribute Age is not represented in this relation. This relation solely consists of simple attributes. The **COMPANION** entity has Name as composite attribute. The simple attributes First\_Name, Last\_Name, and Middle\_Name that compose Name are included in this relation. Moreover, the P\_id, the primary key of the owner entity **CUSTOMER**, is included. P\_id and the partial keys First\_Name, Last\_Name, and Middle\_Name are combined to represent the primary key of this relation.

* 1. **DEPARTMENT**

|  |  |  |
| --- | --- | --- |
| Department\_Name | Branch\_Nb | Description |

The weak entity **DEPARTMENT** contains only simple attributes. The weak entity does not have any derived, multivalued, or composite attributes. Moreover, the Branch\_Nb, the primary of the owner entity **BRANCH**, is included. Branch\_Nb and the partial key Department\_Name are combined to represent the primary key of this relation.

* 1. **EMERGENCY\_CONTACT1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | P\_id | Chat\_id | Relationship |

The **EMERGENCY\_CONTACT1** entity contains simple and composite attributes. This relation solely consists of simple attributes. The **EMERGENCY\_CONTACT1** entity has Name as composite attribute. The simple attributes First\_Name, Last\_Name, and Middle\_Name that compose Name are included in this relation. Moreover, the P\_id, the primary key of the owner entity **CUSTOMER**, is included. P\_id and the partial keys First\_Name, Last\_Name, and Middle\_Name are combined to represent the primary key of this relation.

* 1. **EMERGENCY\_CONTACT2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | essn | Chat\_id | Relationship |

The **EMERGENCY\_CONTACT2** entity contains simple and composite attributes. This relation solely consists of simple attributes. The **EMERGENCY\_CONTACT2** entity has Name as composite attribute. The simple attributes First\_Name, Last\_Name, and Middle\_Name that compose Name are included in this relation. Moreover, the essn, the primary key of the owner entity **EMPLOYEE,** is included. Essn and the partial keys First\_Name, Last\_Name, and Middle\_Name are combined to represent the primary key of this relation.

**STEP 3: Mapping of Binary 1:1 Relation Types**

1. **CONNECTS\_WITH (TICKET)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ticket\_number | Mode\_of\_Transportation | Transport\_company | Baggage\_weight | Seat\_number | Destination\_City | Destination\_State |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Destination\_Country | Destination\_Planet | Departure\_City | Departure\_State | Departure\_Country | Departure\_Planet | Price | Ticket\_type | Is\_booked |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Destination\_Time | Destination\_Date | Departure\_Time | Departure\_Date | Connection\_number |

The **CONNECTS\_WITH** relationship is a self-referencing relationship between two **TICKETs.** Thus, to the **TICKET** table, we add as foreign key the Ticket\_number of the connecting ticket, named Connection\_number.

**STEP 4: Mapping of Binary 1:N Relation Types**

1. **SUPERVISES (EMPLOYEE)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Employee\_ssn | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth | Street\_Number | City | State |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Country | Planet | Gender | Salary | Position | Supervisor\_ssn |

The **SUPERVISES** relationship is a self-referencing relationship that links two **EMPLOYEEs.** Thus, to the **EMPLOYEE** table, we add as foreign key Supervisor\_ssn.

1. **DRAFTS (CONTRACT)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Contract\_id | is\_Refundable | is\_Signed | Expiry\_date | Travel\_reason | Employee\_ssn |

The **DRAFTS** relationship links the **EMPLOYEE** to the **CONTRACT.** Since the **CONTRACT** has the N-side, we add to it, as foreign key, the primary key of **EMPLOYEE -** Employee\_ssn.

1. **WORKS\_FOR (EMPLOYEE)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Employee\_ssn | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth | Street\_Number | City | State |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Country | Planet | Gender | Salary | Position | Supervisor\_ssn | Branch\_nb | Dept\_name |

The **WORKS\_FOR** relationship links the **EMPLOYEE** to **DEPARTMENT.** Thus, Since the **EMPLOYEE** has the N-side, we add to it, as foreign key, the primary key of **DEPARTMENT -** Branch\_nb and Dept\_name combined.

1. **CONTAINS (DEPARTMENT)**

|  |  |  |
| --- | --- | --- |
| Department\_Name | Branch\_Nb | Description |

The **CONTAINS** relationship is the identifying relationship between **BRANCH** and the weak entity type **DEPARTMENT**. Thus, to the **DEPARTMENT** table, we add the primary key of **BRANCH**, Branch\_Nb, as foreign key. However, it has already been added in STEP 2: Mapping of Weak Entity Types, and it is included as part of the primary key of **DEPARTMENT,** along withDepartment\_Name.

1. **BOOKS (TICKET)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ticket\_number | Mode\_of\_Transportation | Transport\_company | Baggage\_weight | Seat\_number | Destination\_City | Destination\_State |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Destination\_Country | Destination\_Planet | Departure\_City | Departure\_State | Departure\_Country | Departure\_Planet | Price | Ticket\_type | Is\_booked |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Destination\_Time | Destination\_Date | Departure\_Time | Departure\_Date | Connection\_number | Booking\_Date | Psprt\_id |

The **BOOKS** relationship links the **CUSTOMER** to the **TICKET.** The Booking\_Date attribute is an attribute for the relationship **BOOKS,** so it is added to the table. Moreover, since the **TICKET** has the N-side, we add to it, as foreign key, the primary key of **CUSTOMER -** Psprt\_nb.

1. **STAYS\_AT (CUSTOMER)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Passport\_id | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Street\_Number | City | State | Country | Planet | Gender | Points\_balance | Stay\_Duration | Hotel\_id |

The **STAYS\_AT** relationship links the **CUSTOMER** to the **HOTEL.** The Stay\_Duration attribute is an attribute for the relationship **STAYS\_AT,** so it is added to the table. Moreover, since the **CUSTOMER** has the N-side, we add to it, as foreign key, the primary key of **CUSTOMER -** Psprt\_nb.

1. **PAYS (BILL)**

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction\_id | Payment\_date | Method\_of\_Payment | Psprt\_id |

The **PAYS** relationship links the **CUSTOMER** to the **BILL.** Since the **BILL** has the N-side, we add to it, as foreign key, the primary key of **CUSTOMER -** Psprt\_nb.

1. **CONTACTS (CUSTOMER)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Passport\_id | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Street\_Number | City | State | Country | Planet | Gender | Points\_balance | Stay\_Duration | Hotel\_id | Brnch\_nb |

The **CONTACT** relationship links the **CUSTOMER** to the **BRANCH.** Since the **CUSTOMER** has the N-side, we add to it as, foreign key, the primary key of **BRANCH -** Brnch\_nb.

1. **PREPARES (ROOM)**

|  |  |  |  |
| --- | --- | --- | --- |
| Room\_Number | H\_id | Capacity | Room\_Type |

The **PREPARES** relationship is the identifying relationship between a **HOTEL** and the weak entity type **ROOM**. Thus, to the **ROOM** table, we add the primary key of **HOTEL** which is H\_id as foreign key. However, it has already been added in STEP 2: Mapping of Weak Entity Types, and it is included as part of the primary key of **ROOM,** along withRoom\_Number.

1. **SIGNS(CONTRACT)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Contract\_id | is\_Refundable | is\_Signed | Expiry\_date | Travel\_reason | Employee\_ssn | Psprt\_id |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ticket\_number | Mode\_of\_Transportation | Transport\_company | Baggage\_weight | Seat\_number | Destination\_City | Destination\_State |

1. **PROVIDES (TICKET)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Destination\_Country | Destination\_Planet | Departure\_City | Departure\_State | Departure\_Country | Departure\_Planet | Price | Ticket\_type | Is\_booked |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination\_Time | Departure\_Time | Connection\_number | Booking\_Date | Psprt\_id | Prtnr\_id |

The **PROVIDES** relationship links the **PARTNER** to the **TICKET.** Since the **TICKET** has the N-side, we add to it, as foreign key, the primary key of **PARTNER -** Prtnr\_id.

1. **Accompanies(Companion)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | P\_id | Chat\_id | Date\_of\_Birth | Gender | Relationship |

The **ACCOMPANIES** relationship is the identifying relationship between a **CUSTOMER** and the weak entity type **COMPANION**. Thus, to the **COMPANION** table, we add the primary key of **CUSTOMER** which is P\_idas foreign key. However, it has already been added in STEP 2: Mapping of Weak Entity Types, and it is included as part of the primary key of **COMPANION,** along withFirst\_Name, Middle\_Name, and Last\_Name.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | P\_id | Chat\_id | Relationship |

1. **HAS\_AN1 (EMERGENCY\_CONTACT1)**

The **HAS\_AN1** relationship is the identifying relationship between a **CUSTOMER** and the weak entity type **EMERGENCY\_CONTACT1**. Thus, to the **EMERGENCY\_CONTACT1** table, we add the primary key of **CUSTOMER,** which is P\_id as foreign key. However, it has already been added in STEP 2: Mapping of Weak Entity Types, and it is included as part of the primary key of **EMERGENCY\_CONTACT1,** along withFirst\_Name, Middle\_Name, and Last\_Name.

1. **HAS\_AN2 (EMERGENCY\_CONTAC2)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | essn | Chat\_id | Relationship |

The **HAS\_AN2** relationship is the identifying relationship between an **EMPLOYEE** and the weak entity type **EMERGENCY\_CONTACT2**. Thus, to the **EMERGENCY\_CONTACT2** table, we add the primary key of **EMPLOYEE,** which is essn as foreign key. However, it has already been added in STEP 2: Mapping of Weak Entity Types, and it is included as part of the primary key of **EMERGENCY\_CONTACT2,** along withFirst\_Name, Middle\_Name, and Last\_Name.

**STEP 5: Mapping of Binary M:N Relation Types**

1. **MANAGES**

|  |  |
| --- | --- |
| Employee\_ssn | Psprt\_id |

The **MANAGES** relationship links the **EMPLOYEE** to the **CUSTOMER**. Since it is of type M:N, we create a new relation (table). The primary key is a combination of the primary keys of **EMPLOYEE** and **CUSTOMER**, namely Employee\_ssn and Psprt\_id.

**STEP 6: Mapping of Multivalued Attributes**

|  |  |
| --- | --- |
| Purchased\_Tickets\_id | Psprt\_id |

1. **CUSTOMER\_PURCHASE\_HISTORY**

The Purchased\_Tickets\_id is a multivalued attribute for the **CUSTOMER**. We create a new relation (table) named **CUSTOMER\_PURCHASE\_HISTORY** for it. The primary key is a combination of the primary key of **CUSTOMER**, namely Psprt\_id and Purchased\_Tickets\_id.

|  |  |
| --- | --- |
| Country | Prtnr\_id |

1. **PARTNER\_OPERATING\_COUNTRIES**

The Operating\_Countries is a multivalued attribute for the **PARTNER**. We create a new relation (table) named **PARTNER\_OPERATING\_COUNTRIES** for it. The primary key is a combination of the primary key of **PARTNER**, namely Prtnr\_id, and Country.

**STEP 7: Mapping of N-ary Relationships**

|  |  |  |
| --- | --- | --- |
| Employee\_ssn | Psprt\_id | Htl\_id |

1. **RECOMMENDS**

The **RECOMMENDS** relationship is a ternary relationship linking the **EMPLOYEE** to the **HOTEL** to the **CUSTOMER**. Thus, we create a new relation (table) for it. The primary key is a combination of the primary keys of **EMPLOYEE**, **HOTEL**, and **CUSTOMER**, namely Employee\_ssn, Psprt\_id, and Htl\_id.

1. **Mapping Result**
   1. **Customer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Passport\_id | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Street\_Number | City | State | Country | Planet | Gender | Points\_balance | Stay\_Duration | Hotel\_id | Brnch\_nb |

1. **Employee**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Employee\_ssn | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth | Street\_Number | City | State |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Country | Planet | Gender | Salary | Position | Supervisor\_ssn | Branch\_nb | Dept\_name |

1. **Ticket**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ticket\_number | Mode\_of\_Transportation | Transport\_company | Baggage\_weight | Seat\_number | Destination\_City | Destination\_State |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Destination\_Country | Destination\_Planet | Departure\_City | Departure\_State | Departure\_Country | Departure\_Planet | Price | Ticket\_type | Is\_booked |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination\_Time | Departure\_Time | Connection\_number | Booking\_Date | Psprt\_id | Prtnr\_id |

1. **Hotel**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Hotel\_id | Hotel\_name | Street\_Number | City | State | Country | Planet |

|  |  |  |  |
| --- | --- | --- | --- |
| Star\_rating | Number\_of\_Meals | Price | is\_Partner |

1. **Room**

|  |  |  |  |
| --- | --- | --- | --- |
| Room\_Number | H\_id | Capacity | Room\_Type |

1. **Contract**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Contract\_id | is\_Refundable | is\_Signed | Expiry\_date | Travel\_reason | Employee\_ssn | Psprt\_id |

1. **Companion**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | P\_id | Chat\_id | Date\_of\_Birth | Gender | Relationship |

1. **Bill**

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction\_id | Payment\_date | Method\_of\_Payment | Psprt\_id |

1. **Department**

|  |  |  |
| --- | --- | --- |
| Department\_Name | Branch\_Nb | Description |

1. **Emergency\_Contact1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | P\_id | Chat\_id | Relationship |

1. **Emergency\_Contact2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | essn | Chat\_id | Relationship |

1. **Branch**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Branch\_Number | Street\_Number | City | State | Country | Planet |

1. **Partner**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Partner\_id | Name | City | State | Country | Planet | Service | Street\_Number |

1. **Customer\_Purchase\_History**

|  |  |
| --- | --- |
| Purchased\_Tickets\_id | Psprt\_id |

1. **Partner\_Operating\_Countries**

|  |  |
| --- | --- |
| Country | Prtnr\_id |

1. **Recommends**

|  |  |  |
| --- | --- | --- |
| Employee\_ssn | Psprt\_id | Htl\_id |

|  |  |
| --- | --- |
| Employee\_ssn | Psprt\_id |

1. **Manages**
2. **SQL Queries and Oracle Server**

After designing the ER diagram for Metatravel and mapping this diagram into relational database design, it is time to start creating the tables for our database on the Oracle Database Server.

1. **Customer:**

**CREATE** **TABLE** CUSTOMER

**(**

Passport\_id **CHAR(**9**) PRIMARY KEY ,**

First\_Name **VARCHAR(**15**) NOT NULL ,**

Middle\_Name **VARCHAR(**15**) NOT NULL ,**

Last\_Name **VARCHAR(**15**) NOT NULL,**

Chat\_id **VARCHAR(**8**) NOT NULL ,**

Date\_of\_Birth **DATE NOT NULL ,**

Street\_Number **VARCHAR(**5**) ,**

City **VARCHAR**(10**) ,**

State **VARCHAR**(10**) ,**

Country **VARCHAR**(15**) ,**

Planet **VARCHAR**(15**) ,**

Gender **CHAR(**1**)CHECK(**GENDER **IN(**'M'**,**'F'**)) ,**

Points\_balance **INT** **NOT NULL ,**

Hotel\_id **CHAR(**9**) ,**

Brnch\_nb **VARCHAR(**5**) ,**

**FOREIGN KEY**(Hotel\_id) **REFERENCES** HOTEL**(**HOTEL\_ID**) ,**

**FOREIGN KEY**(Brnch\_nb) **REFERENCES** BRANCH**(**BRANCH\_NUMBER**)**

**);**

1. **Employee:**

**CREATE TABLE** EMPLOYEE

**(**

Employee\_ssn **CHAR(9) PRIMARY KEY,**

First\_Name **VARCHAR(15) NOT NULL ,**

Middle\_Name **VARCHAR(15) NOT NULL ,**

Last\_Name **VARCHAR(15) NOT NULL,**

Chat\_id **VARCHAR(8) NOT NULL ,**

Date\_of\_Birth **DATE NOT NULL ,**

Street\_Number **VARCHAR(5) ,**

City **VARCHAR(10) ,**

State **VARCHAR(10) ,**

Country **VARCHAR(15) ,**

Planet **VARCHAR(15) ,**

Gender **CHAR(1)CHECK(**GENDER **IN(**'M'**,**'F'**)),**

Salary **NUMERIC(6,2),**

Position **VARCHAR(30),**

Supervisor\_ssn **CHAR(9),**

Branch\_num **VARCHAR(5),**

**FOREIGN KEY(**Supervisor\_ssn**) REFERENCES** EMPLOYEE(EMPLOYEE\_SSN**),**

**FOREIGN KEY(**Branch\_num**) REFERENCES** BRANCH**(**BRANCH\_NUMBER**)**

**);**

1. **Ticket:**

**CREATE TABLE** TICKET

**(**

Ticket\_Number **CHAR(7) PRIMARY KEY ,**

Mode\_Of\_Transportation **VARCHAR(20) NOT NULL ,**

Transport\_Company **VARCHAR(20) NOT NULL ,**

Baggage\_Weight **NUMERIC(2**,**2) ,**

Seat\_Number **INT NOT NULL ,**

Destination\_City **VARCHAR(10) NOT NULL,**

Destination\_State **VARCHAR(10) NOT NULL,**

Destination\_Country **VARCHAR(15) NOT NULL ,**

Destination\_Planet **VARCHAR(15) NOT NULL,**

Destination\_Time **TIMESTAMP NOT NULL,**

Price **NUMERIC(3**,**2) NOT NULL ,**

Ticket\_Type **VARCHAR(20) CHECK**(TICKET\_TYPE **IN(**'One-Way'**,**'Round- T Trip'**)) NOT NULL ,**

Is\_Booked **NUMBER(1)** **NOT** **NULL** CHECK(Is\_Booked **IN**(0,1))**,**

Departure\_City **VARCHAR(10) NOT NULL,**

Departure\_State **VARCHAR(10) NOT NULL,**

Departure\_Country **VARCHAR(15) NOT NULL,**

Departure\_Planet **VARCHAR(15) NOT NULL,**

Departure\_Time **TIMESTAMP NOT NULL,**

Connection\_Number **CHAR(7)**,

Psprt\_id **CHAR(9),**

Prtnr\_id **CHAR(9),**

Booking\_Date **DATE,**

**FOREIGN KEY(**Connection\_Number**)** **REFERENCES** TICKET**(**Ticket\_Number**),**

**FOREIGN KEY(**Psprt\_id**) REFERENCES** CUSTOMER**(**Passport\_id**),**

**FOREIGN KEY(**Prtnr\_id**)** **REFERENCES** PARTNER**(**Partner\_id**)** **);**

1. **Hotel:**

**CREATE TABLE** HOTEL

(

HOTEL\_ID **CHAR(9)** **PRIMARY KEY** ,

HOTEL\_NAME **VARCHAR(20)** **NOT NULL**,

STREET\_NUMBER **VARCHAR(5) ,**

CITY **VARCHAR(10) ,**

STATE **VARCHAR(10) ,**

COUNTRY **VARCHAR(15) ,**

PLANET **VARCHAR(15) ,**

STAR\_RATING **VARCHAR(1),**

NUMBER\_OF\_MEALS **VARCHAR(4),**

PRICE **VARCHAR**(**5**) ,

IS\_PARTNER **CHAR**(**1**) **CHECK**(IS\_PARTNER **IN** **(**'Y'**,**'N'**))**

);

1. **Room:**

**CREATE** **TABLE** ROOM

(

H\_id **CHAR(9) ,**

Room\_Number **CHAR(3)** **,**

Capacity **INT ,**

Room\_Type **VARCHAR(15)** **CHECK**(ROOM\_TYPE **IN(**'Single B B Bed'**,**'Double Bed'**,**'Triple Bed'**,**'King-Sized Bed'**,**'Queen- S S Sized Bed'**,**'Suite'**)) ,**

**FOREIGN KEY** (H\_id) **REFERENCES** HOTEL(HOTEL\_ID),

**PRIMARY KEY(**H\_id**,** Room\_Number**)**

**);**

1. **Contract:**

**CREATE TABLE** CONTRACT

**(**

CONTRACT\_ID **CHAR(7)** PRIMARY KEY**,**

is\_Refundable **CHAR(1)** **CHECK**(is\_Refundable **IN** **(**'Y'**,**'N'**)) NOT NULL,**

is\_Signed **CHAR(1) CHECK**(is\_Signed **IN** **(**'Y','N')**)** **NOT NULL**,

Expiry\_Date **Date** **NOT NULL**,

Travel\_Reason **VARCHAR(**20**) NOT NULL**,

essn **CHAR(9)** **NOT NULL**,

Psprt\_id **CHAR(9)**,

**FOREIGN KEY** (essn) **REFERENCES** EMPLOYEE(Employee\_ssn),

**FOREIGN KEY (**Psprt\_id) **REFERENCES** CUSTOMER(Passport\_id)

**);**

1. **Companion:**

**CREATE TABLE** COMPANION

**(**

First\_Name **VARCHAR(15)** **NOT NULL ,**

Middle\_Name **VARCHAR(15) NOT NULL,**

Last\_Name **VARCHAR(15)** **NOT NULL,**

P\_id **CHAR(9) ,**

Chat\_id **VARCHAR(8)** **NOT NULL ,**

Date\_Of\_Birth **DATE ,**

Gender CHAR(1) **CHECK**(GENDER **IN(**'M'**,**'F'**)) ,**

Relationship **VARCHAR(20)** **NOT NULL ,**

**FOREIGN KEY** (P\_id) **REFERENCES** CUSTOMER(Passport\_id) **,**

**PRIMARY KEY(**First\_Name**,**Middle\_Name**,**Last\_Name**,**P\_id**)**

**);**

1. **Bill:**

**CREATE TABLE** BILL

**(**

Transaction\_id **VARCHAR(15) PRIMARY KEY ,**

Payment\_date **DATE** **NOT NULL ,**

Method\_Of\_Payment **VARCHAR(20) NOT NULL ,**

Psprt\_id **CHAR(9) ,**

**FOREIGN KEY**(Psprt\_id)**REFERENCES** CUSTOMER**(**Passport\_id**)**

**);**

1. **Department:**

**CREATE TABLE** DEPARTMENT

**(**

Branch\_Nb **VARCHAR(5)** **NOT NULL ,**

Department\_Name **VARCHAR(30) NOT NULL ,**

Description **VARCHAR(100) NOT NULL ,**

**FOREIGN KEY** **(**Branch\_Nb**)** **REFERENCES** BRANCH**(**BRANCH\_NUMBER**) ,**

**PRIMARY KEY(**Branch\_Nb,Department\_Name**)**

**);**

1. **EMERGENCY\_CONTACT1**

**CREATE** **TABLE** EMERGENCY\_CONTACT1

**(**

First\_Name **VARCHAR(**15**) NOT NULL,**

Middle\_Name **VARCHAR(**15**),**

Last\_Name **VARCHAR(**15**) NOT NULL,**

Chat\_id **VARCHAR(**8**) NOT NULL,**

Relatioship **VARCHAR**(10**),**

p\_id CHAR(9),

**PRIMARY KEY(**First\_Name, Middle\_Name, Last\_Name, p\_id**),**

**FOREIGN KEY**(p\_id) **REFERENCES** CUSTOMER**(**Passport\_id**)**

**);**

1. **EMERGENCY\_CONTACT2**

**CREATE** **TABLE** EMERGENCY\_CONTACT2

**(**

First\_Name **VARCHAR(**15**) NOT NULL,**

Middle\_Name **VARCHAR(**15**),**

Last\_Name **VARCHAR(**15**) NOT NULL,**

Chat\_id **VARCHAR(**8**) NOT NULL,**

Relatioship **VARCHAR**(10**),**

essn CHAR(9),

**PRIMARY KEY(**First\_Name, Middle\_Name, Last\_Name, essn**),**

**FOREIGN KEY**(essn) **REFERENCES** EMPLOYEE**(**[EMPLOYEE\_SSN](javascript:ret_Column('EMPLOYEE_SSN');)**)**

**);**

1. **BRANCH**

**CREATE TABLE** BRANCH

(

Branch\_Number **VARCHAR(5)PRIMARY KEY,**

Street\_Number **VARCHAR(5),**

City **VARCHAR(10),**

State **VARCHAR(10),**

Country **VARCHAR(15),**

Planet **VARCHAR(15)**

);

1. **PARTNER**

**CREATE** **TABLE** PARTNER

**(**

Partner\_id **CHAR(**9**) PRIMARY KEY,**

Name **VARCHAR(**20**) NOT NULL,**

City **VARCHAR**(15**),**

State **VARCHAR**(10**),**

Country **VARCHAR**(15**),**

Planet **VARCHAR**(15**),**

Service **VARCHAR**(15**) NOT NULL,**

Street\_Number **VARCHAR(**6**)**

**);**

1. **CUSTOMER\_PURCHASE\_HISTORY**

**CREATE** **TABLE** CUSTOMER\_PURCHASED\_HISTORY

(

Purchase\_Tickets\_id **CHAR(**9),

Psprt\_id CHAR(9),

PRIMARY KEY (Purchase\_Tickets\_id, Psprt\_id),

FOREIGN KEY(Purchase\_Tickets\_id) REFERENCES TICKET(Ticket\_Number),

FOREIGN KEY(Psprt\_id) REFERENCES CUSTOMER(Passport\_id)

);

1. **PARTNER\_OPERATING\_COUNTRIES**

**CREATE** **TABLE** PARTNER\_OPERATING\_COUNTRIES

(

Country **VARCHAR(**15**)** ,

Prtnr\_id **CHAR**(9),

PRIMARY KEY (Country ,Prtnr\_id),

FOREIGN KEY(Prtnr\_id) REFERENCES PARTNER(Partner\_id)

);

1. **RECOMMENDS**

**CREATE** **TABLE** RECOMMENDS

(

Employee\_ssn CHAR(9),

Psprt\_id CHAR(9),

Htl\_id CHAR(9),

PRIMARY KEY (Employee\_ssn,psprt\_id,htl\_id),

FOREIGN KEY(Employee\_ssn) REFERENCES EMPLOYEE(Employee\_ssn),

FOREIGN KEY(Psprt\_id) REFERENCES CUSTOMER(Passport\_id),

FOREIGN KEY(Htl\_id) REFERENCES HOTEL(Hotel\_id)

);

1. **MANAGES**

**CREATE** **TABLE** MANAGES

(

Employee\_ssn CHAR(9),

Psprt\_id CHAR(9),

PRIMARY KEY (Employee\_ssn,psprt\_id),

FOREIGN KEY(Employee\_ssn) REFERENCES EMPLOYEE(Employee\_ssn),

FOREIGN KEY(Psprt\_id) REFERENCES CUSTOMER(Passport\_id));

1. **Table Descriptions**

After creating the tables on the Oracle Database Server, we are going to list the names of all our entities. Then, we are going to view description of each table created.

**List of Tables: Query + Result**

* **Query:**

**SELECT** DISTINCT OBJECT\_NAME

**FROM** USER\_OBJECTS

**WHERE** OBJECT\_TYPE = 'TABLE';

* **Result:**

**OBJECT\_NAME**

**----------------------------------------------------------------------------------------------------------------**

CUSTOMER

EMPLOYEE

TICKET

HOTEL

ROOM

COMPANION

BILL

DEPARTMENT

EMERGENCY\_CONTACT

EMERGENCY\_CONTACT2

BRANCH

PARTNER

CUSTOMER\_PURCHASE\_HISTORY

RECOMENDS

PARTNER\_OPERATING\_COUNTRIES

1. **CUSTOMER Description:**

SQL> DESC CUSTOMER

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| PASSPORT\_ID [P1] | CHAR(9) | No |
| FIRST\_NAME | VARCHAR2(15) | No |
| MIDDLE\_NAME | VARCHAR2(15) | No |
| LAST\_NAME | VARCHAR2(15) | No |
| CHAT\_ID | VARCHAR2(8) | No |
| DATE\_OF\_BIRTH | DATE | No |
| STREET\_NUMBER | VARCHAR2(5) | Yes |
| CITY | VARCHAR2(15) | Yes |
| STATE | VARCHAR2(15) | Yes |
| COUNTRY | VARCHAR2(15) | Yes |
| PLANET | VARCHAR2(15) | Yes |
| GENDER | CHAR(1) | Yes |
| POINTS\_BALANCE | NUMBER | No |
| HOTEL\_ID | CHAR(9) | Yes |
| BRNCH\_NB | VARCHAR2(5) | Yes |

1. **EMPLOYEE Description:**

SQL>DESC EMPLOYEE

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| EMPLOYEE\_SSN [P1] | CHAR(9) | No |
| FIRST\_NAME | VARCHAR2(15) | No |
| MIDDLE\_NAME | VARCHAR2(15) | No |
| LAST\_NAME | VARCHAR2(15) | No |
| CHAT\_ID | VARCHAR2(8) | No |
| DATE\_OF\_BIRTH | DATE | No |
| STREET\_NUMBER | VARCHAR2(5) | Yes |
| CITY | VARCHAR2(15) | Yes |
| STATE | VARCHAR2(15) | Yes |
| COUNTRY | VARCHAR2(15) | Yes |
| PLANET | VARCHAR2(15) | Yes |
| GENDER | CHAR(1) | Yes |
| SALARY | NUMBER(6,2) | Yes |
| POSITION | VARCHAR2(30) | Yes |
| SUPERVISOR\_SSN | CHAR(9) | Yes |
| BRANCH\_NUM | VARCHAR2(5) | Yes |

1. **TICKET Description:**

SQL>DESC TICKET

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| TICKET\_NUMBER | CHAR(7) | No |
| MODE\_OF\_TRANSPORTATION | VARCHAR2(20) | No |
| TRANSPORT\_COMPANY | VARCHAR2(20) | No |
| BAGGAGE\_WEIGHT | NUMBER(4,2) | Yes |
| SEAT\_NUMBER | NUMBER | No |
| DESTINATION\_CITY | VARCHAR2(10) | No |
| DESTINATION\_STATE | VARCHAR2(10) | No |
| DESTINATION\_COUNTRY | VARCHAR2(15) | No |
| DESTINATION\_PLANET | VARCHAR2(15) | No |
| DESTINATION\_TIME | TIMESTAMP(6) | No |
| PRICE | NUMBER(3,2) | No |
| TICKET\_TYPE | VARCHAR2(20) | No |
| IS\_BOOKED | NUMBER(1,0) | No |
| DEPARTURE\_CITY | VARCHAR2(10) | No |
| DEPARTURE\_STATE | VARCHAR2(10) | No |
| DEPARTURE\_COUNTRY | VARCHAR2(15) | No |
| DEPARTURE\_PLANET | VARCHAR2(15) | No |
| DEPARTURE\_TIME | TIMESTAMP(6) | No |
| CONNECTION\_NUMBER | CHAR(7) | Yes |
| PSPRT\_ID | CHAR(9) | Yes |
| PRTNR\_ID | CHAR(9) | Yes |
| Booking\_date | DATE | Yes |

1. **HOTEL Description:**

SQL>DESC HOTEL

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| HOTEL\_ID [p1] | CHAR(9) | No |
| HOTEL\_NAME | VARCHAR2(20) | No |
| STREET\_NUMBER | VARCHAR2(5) | Yes |
| CITY | VARCHAR2(10) | Yes |
| STATE | VARCHAR2(10) | Yes |
| COUNTRY | VARCHAR2(15) | Yes |
| PLANET | VARCHAR2(15) | Yes |
| STAR\_RATING | VARCHAR2(1) | Yes |
| NUMBER\_OF\_MEALS | VARCHAR2(4) | Yes |
| PRICE | VARCHAR2(5) | Yes |
| IS\_PARTNER | CHAR(1) | Yes |

1. **ROOM Description:**

SQL>DESC ROOM

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| H\_ID | CHAR(9) | No |
| ROOM\_NUMBER | CHAR(3) | No |
| CAPACITY | NUMBER | Yes |
| ROOM\_TYPE | VARCHAR2(15) | Yes |

1. **CONTRACT Description:**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| CONTRACT\_ID | CHAR(7) | No |
| IS\_REFUNDABLE | CHAR(1) | No |
| IS\_SIGNED | CHAR(1) | No |
| EXPIRY\_DATE | DATE | No |
| TRAVEL\_REASON | VARCHAR2(20) | No |
| ESSN | CHAR(9) | No |
| PSPRT\_ID | CHAR(9) | Yes |

1. **COMPANION Description:**

SQL>DESC COMPANION

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| FIRST\_NAME [P1] | VARCHAR2(15) | No |
| MIDDLE\_NAME [P2] | VARCHAR2(15) | No |
| LAST\_NAME [P3] | VARCHAR2(15) | No |
| P\_ID [P4] | CHAR(9) | No |
| CHAT\_ID | VARCHAR2(8) | No |
| DATE\_OF\_BIRTH | DATE | Yes |
| GENDER | CHAR(1) | Yes |
| RELATIONSHIP | VARCHAR2(20) | No |

1. **BILL Description:**

SQL>DESC BILL

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| TRANSACTION\_ID [P1] | VARCHAR2(15) | No |
| PAYMENT\_DATE | DATE | No |
| METHOD\_OF\_PAYMENT | VARCHAR2(20) | No |
| PSPRT\_ID | CHAR(9) | Yes |

1. **DEPARTMENT Description**

SQL>DESC DEPARTMENT

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| BRANCH\_NB [p1] | VARCHAR2(5) | No |
| DEPARTMENT\_NAME [P2] | VARCHAR2(30) | No |
| DESCRIPTION | VARCHAR2(100) | No |

1. **EMERGENCY CONTACT1 Description:**

SQL>DESC EMERGENCY\_CONTACT

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| FIRST\_NAME [P1] | VARCHAR2(15) | No |
| MIDDLE\_NAME [P2] | VARCHAR2(15) | Yes |
| LAST\_NAME [P3] | VARCHAR2(15) | No |
| CHAT\_ID | VARCHAR2(8) | No |
| RELATIOSHIP | VARCHAR2(10) | Yes |
| P\_ID [P4] | CHAR(9) | Yes |

1. **EMERGENCY CONTACT2 Description:**

SQL>DESC EMERGENCY\_CONTACT2

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| FIRST\_NAME [P1] | VARCHAR2(15) | No |
| MIDDLE\_NAME [P2] | VARCHAR2(15) | Yes |
| LAST\_NAME [P3] | VARCHAR2(15) | No |
| CHAT\_ID | VARCHAR2(8) | No |
| RELATIOSHIP | VARCHAR2(10) | Yes |
| ESSN [P4] | CHAR(9) | Yes |

1. **BRANCH Description:**

SQL>DESC BRANCH

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| BRANCH\_NUMBER [P1] | VARCHAR2(5) | No |
| STREET\_NUMBER | VARCHAR2(5) | Yes |
| CITY | VARCHAR2(10) | Yes |
| STATE | VARCHAR2(10) | Yes |
| COUNTRY | VARCHAR2(15) | Yes |
| PLANET | VARCHAR2(15) | Yes |

1. **PARTNER Description:**

SQL>DESC PARTNER

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| PARTNER\_ID [P1] | CHAR(9) | No |
| NAME | VARCHAR2(20) | No |
| CITY | VARCHAR2(15) | Yes |
| STATE | VARCHAR2(10) | Yes |
| COUNTRY | VARCHAR2(15) | Yes |
| PLANET | VARCHAR2(15) | Yes |
| SERVICE | VARCHAR2(15) | No |
| STREET\_NUMBER | VARCHAR2(6) | Yes |

1. **CUSTOMER PURCHASE HISTORY Description:**

SQL>DESC CUSTOMER\_PURCHASE\_HISTORY

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| PURCHASE\_TICKETS\_ID [P1] | CHAR(9) | No |
| PSPRT\_ID [P2] | CHAR(9) | No |

1. **PARTNER OPERATING COUNTRIES Description:**

SQL>DESC PARTNER\_OPERATING\_COUNTRIES

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| COUNTRY [P1] | VARCHAR2(20) | No |
| PRTNR\_ID [P2] | CHAR(9) | No |

1. **RECOMMENDS Description:**

SQL>DESC RECOMMENDS

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| EMPLOYEE\_SSN [P1] | CHAR(9) | No |
| PSPRT\_ID [P2] | CHAR(9) | No |
| HTL\_ID [P3] | CHAR(9) | No |

1. **MANAGES Description:**

SQL>DESC RECOMMENDS

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Nullable |
| EMPLOYEE\_SSN | CHAR(9) | No |
| PSPRT\_ID | CHAR(9) | No |

1. **Inserting Data**
   1. **CUSTOMER:**
2. **INSERT INTO CUSTOMER VALUES('127362306', 'Peter', 'Austin', 'McCallister', '85764399', TO\_DATE('1978-03-07','YYYY-MM-DD'), '** **60093', 'Chicago', 'Illinois', 'United States', 'Earth', 'M', 4, '754125479', '00001');**
3. **INSERT INTO CUSTOMER VALUES('781457987', 'Harry', 'James', 'Potter', '14787614', TO\_DATE('1980-07-31','YYYY-MM-DD'),'20741', 'London', NULL, 'England', 'Earth', 'M', 2, '125227690', '00016');**
4. **INSERT INTO CUSTOMER VALUES('984230491', 'Pete', 'Michael', 'Davidson', '72838172', TO\_DATE('1993-11-16','YYYY-MM-DD'), '62549', 'Staten Island', 'New York', 'United States', 'Earth', 'M', 12, '620390011', '00030');**
5. **INSERT INTO CUSTOMER VALUES('721230456', 'Thomas', 'Stanley', 'Holland', '29145713', TO\_DATE('1996-06-01','YYYY-MM-DD'), '62549', 'Kingston', DEFAULT, 'England', 'Earth', 'M', 10, '620390011', '00030');**
6. **INSERT INTO CUSTOMER VALUES('644578149', 'Ryan', 'Rodney', 'Reynolds', '23781639', TO\_DATE('1976-10-23','YYYY-MM-DD'), '58172', 'Vancouver', DEFAULT, 'Canada', 'Earth', 'M', 8, '620390011', '00030');**
7. **INSERT INTO CUSTOMER VALUES('432779156', 'Michael', 'Lee', 'Taylor', '81928265', TO\_DATE('1960-04-04','YYYY-MM-DD'), '79144', 'Boston', 'Massachusetts', 'United States', 'Earth', 'M', 14, NULL, '00088');**
8. **INSERT INTO CUSTOMER VALUES('432779157', 'Peter', 'Matthew', 'Taylor', '81928266', TO\_DATE('1992-05-23','YYYY-MM-DD'), '79144', 'Boston', 'Massachusetts', 'United States', 'Earth', 'M', 14, NULL, '00088');**
9. **INSERT INTO CUSTOMER VALUES('601928374', 'Anton', 'Oleg', 'Lavrov', '81928267', TO\_DATE('1991-04-23','YYYY-MM-DD'), '79144', 'Boston', NULL, 'Moscow', 'Earth', 'M', 13, NULL, '00066');**
10. **INSERT INTO CUSTOMER VALUES('601928375', 'Ilya', 'Aleksandr', 'Pavlov', '81928268', TO\_DATE('1992-05-18','YYYY-MM-DD'), '**79144**', 'Moscow', NULL, 'Moscow', 'Earth', 'M', 7, NULL, '00066');**
11. **INSERT INTO CUSTOMER VALUES('700192018', 'Andrea', 'Sofia', 'Rossi', '81920918', TO\_DATE('2001-09-21', 'YYYY-MM-DD'), '84521', 'Milan', DEFAULT, 'Italy', 'Earth', 'F', 20, '720901923', '00044');**
12. **INSERT INTO CUSTOMER VALUES('149823934', 'Kumar', 'Aarav' , 'Patil', '13457309', TO\_DATE('1998-02-25', 'YYYY-MM-DD'), '15612', 'New Delhi', DEFAULT, 'India', 'Earth', 'M', 100, NULL, '00099');**
13. **INSERT INTO CUSTOMER VALUES('556891162', 'Ibrahim', 'Rabih' , 'Ammar', '37810825', TO\_DATE('1995-04-19', 'YYYY-MM-DD'), '67138', 'Beirut', DEFAULT, 'Lebanon', 'Earth', 'M', 41, '43023817', '00022');**
    1. **EMPLOYEE:**
14. **INSERT INTO EMPLOYEE VALUES('889928456', 'James', 'Harold', 'Maddison', '71817135', TO\_DATE('1989-07-15','YYYY-MM-DD'),'61728', 'Chicago', 'Illinois', 'United States', 'Earth', 'M', 6.02, 'Travel Agent', 61229982, '00001');**
15. **INSERT INTO EMPLOYEE VALUES('61229982', 'John', 'Michael', 'Smith', '71817123', TO\_DATE('1986-04-25','YYYY-MM-DD'), '** **61721', 'Chicago', 'Illinois', 'United States', 'Earth', 'M', 7.02, 'Manager', DEFAULT, '00001');**

1. **INSERT INTO EMPLOYEE VALUES('439088731', 'Simon', 'Ethan', 'Williams', '71817119', TO\_DATE('1988-08-11','YYYY-MM-DD'),'43128', 'London', DEFAULT, 'England', 'Earth', 'M', 6.02, 'Travel Agent', 420885610, '00016');**
2. **INSERT INTO EMPLOYEE VALUES('420885610', 'Faith', 'Lena', 'Allen', '71817578', TO\_DATE('** **1985-01-12','YYYY-MM-DD'),'43131', 'London', DEFAULT, 'England', 'Earth', 'F', 7.02, 'Manager', DEFAULT, '00016');**
3. **INSERT INTO EMPLOYEE VALUES('439088732', 'Elizabeth', 'Olivia', 'Coleman', '63391035', TO\_DATE('1985-05-13','YYYY-MM-DD'),'91023', 'Fort Worth', 'Texas', 'United States', 'Earth', 'F', 6.02, 'Travel Agent', '910284334', '00030');**
4. **INSERT INTO EMPLOYEE VALUES('910284334', 'Grace', 'Amelia', 'Robert', '81093581', TO\_DATE('** **1980-08-19','YYYY-MM-DD'),'91029', 'Fort Worth', 'Texas', 'United States', 'Earth', 'F', 7.02, 'Manager', DEFAULT, '00030');**
5. **INSERT INTO EMPLOYEE VALUES('819026609', 'Akira', 'Hiroshi', 'Kiyama', '81093582', TO\_DATE('** **1979-01-02','YYYY-MM-DD'),'17284',** **'Osaka', NULL, 'Japan', 'Earth', 'M', 6.02, 'Travel Agent', 819026608, '00088');**
6. **INSERT INTO EMPLOYEE VALUES('819026608', 'Izumi', 'Sora', 'Sakai', '81093583', TO\_DATE('** **1976-03-19','YYYY-MM-DD'),'45665', 'Osaka', NULL, 'Japan', 'Earth', 'F', 7.02, 'Manager', DEFAULT, '00088');**
7. **INSERT INTO EMPLOYEE VALUES('611209182', 'Blessing', 'Abigail', 'Egbe', '65128090', TO\_DATE('** **1974-02-19','YYYY-MM-DD'),'56710', 'Lagos', NULL, 'Nigeria', 'Earth', 'F', 8.02, 'Researcher', '611209181', '00055');**
8. **INSERT INTO EMPLOYEE VALUES('611209181', 'Chinara', 'Abebi', 'Okoye', '65128090', TO\_DATE('** **1970-10-29','YYYY-MM-DD'),'61092', 'Lagos', NULL, 'Nigeria', 'Earth', 'F', 9.02, 'Research Manager', DEFAULT, '00055');**
9. **INSERT INTO EMPLOYEE VALUES('344309012', 'Natasha', 'Anastasia', 'Smirnov', '65128091', TO\_DATE('** **1997-11-26','YYYY-MM-DD'),'33801', 'Moscow', NULL, 'Russia', 'Earth', 'F', 5.02, 'Travel Agent Manager', NULL, '00066');**
10. **INSERT INTO EMPLOYEE VALUES('344309013', 'Ivan', 'Dimitri', 'Orlov', '65128092', TO\_DATE('1995-12-03','YYYY-MM-DD'),'61092', 'Moscow', NULL, 'Russia', 'Earth', 'M', 7.02, 'HR Manager', DEFAULT, '00066');**
11. **INSERT INTO EMPLOYEE VALUES(****'301928371', 'Eduardo', 'Antonio', 'Bruno', '99817134', TO\_DATE('1990-08-05'), '56527', 'Milan', DEFAULT, 'Italy', 'Earth', 'M', 6.02, 'Travel Agent', '301928372', '00044');**
12. **INSERT INTO EMPLOYEE VALUES('301928372', 'Alice', 'Chiara', 'Greco', '99817133', TO\_DATE('1990-08-05', 'YYYY-MM-DD'), '56527', 'Milan', DEFAULT, 'Italy', 'Earth', 'F', 7.02, 'Travel Agent Manager', DEFAULT, '00044');**
13. **INSERT INTO EMPLOYEE VALUES('4****91022860', 'Viraj', 'Rohan', 'Sharma', '99817134', TO\_DATE('1991-03-11', 'YYYY-MM-DD'), '56528', 'New Delhi', DEFAULT, 'India', 'Earth', 'M', 6.02, 'Travel Agent', '301928372', '00099');**
14. **INSERT INTO EMPLOYEE VALUES('491022861', 'Kiaan', 'Advik', 'Singh', '99817135', TO\_DATE('1988-08-06', 'YYYY-MM-DD'), '56529', 'New Delhi', DEFAULT, 'India', 'Earth', 'M', 7.02, 'Travel Agent Manager', DEFAULT, '00099');**
15. **INSERT INTO EMPLOYEE VALUES('801927541', 'Ghada', 'Maria', 'Saad', '99817136', TO\_DATE('1986-03-06', 'YYYY-MM-DD'), '56320', 'Beirut', DEFAULT, 'Lebanon', 'Earth', 'F', 6.02, 'Travel Agent','801927542', '00022');**
16. **INSERT INTO EMPLOYEE VALUES('801927542', 'Sarah', 'Hala', 'Mrad', '99817137', TO\_DATE('1981-08-18', 'YYYY-MM-DD'), '** **56457', 'Beirut', DEFAULT, 'Lebanon', 'Earth', 'F', 7.02, 'Travel Agent Manager', DEFAULT, '00022');**
    1. **TICKET:**
17. **INSERT INTO TICKET VALUES('1457855','Plane', 'American Airlines', 64, 58, 'New York', 'New York', 'United States', 'Earth', TO\_TIMESTAMP('2021-12-24 10:05:00', 'YYYY-MM-DD HH:MI:SS'),0.07, 'Round-Trip', 1, 'Chicago', 'Illinois' , 'United States', 'Earth', TO\_TIMESTAMP('2021-12-24 8:05:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '127362306', '376180927', TO\_DATE('2021-11-24', 'YYYY-MM-DD'));**
18. **INSERT INTO TICKET VALUES('1588511', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', TO\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '781457987',' 541180927', TO\_DATE('2021-09-04', 'YYYY-MM-DD'));**
19. **INSERT INTO TICKET VALUES('1588512', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '781457987',' 541180927', TO\_DATE('2021-09-04', 'YYYY-MM-DD'));**
20. **INSERT INTO TICKET VALUES('1588513', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', 'TO\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '781457987',' 541180927', TO\_DATE('2021-09-04', 'YYYY-MM-DD'));**
21. **INSERT INTO TICKET VALUES('1588514', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '781457987',' 541180927', TO\_DATE('2021-09-04', 'YYYY-MM-DD'));**
22. **INSERT INTO TICKET VALUES('1588515', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '781457987',' 541180927', TO\_DATE('2021-09-04', 'YYYY-MM-DD'));**
23. **INSERT INTO TICKET VALUES('1588516', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '781457987',' 541180927', TO\_DATE('2021-09-04', 'YYYY-MM-DD'));**
24. **INSERT INTO TICKET VALUES('1288511', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, NULL,' 541180927', NULL)**
25. **INSERT INTO TICKET VALUES('1288512', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, NULL,' 541180927', NULL)**
26. **INSERT INTO TICKET VALUES('1288513', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, NULL,' 541180927', NULL)**
27. **INSERT INTO TICKET VALUES('1288514', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, NULL,' 541180927', NULL)**
28. **INSERT INTO TICKET VALUES('1288515', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, NULL,' 541180927', NULL)**
29. **INSERT INTO TICKET VALUES('1288516', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, NULL,' 541180927', NULL)**
30. **INSERT INTO TICKET VALUES('1288517', 'Train', 'Hogwarts Express', 40, 45, 'Hogwarts', DEFAULT, 'Scotland', 'Earth', To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.01, 'One-Way', 1, 'London', DEFAULT, 'England', 'Earth', TO\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, NULL,' 541180927', NULL)**

1. **INSERT INTO TICKET VALUES('8192002', 'Spaceship', 'SpaceX', 60, 67, 'Nuwa', DEFAULT, 'Tempe Mensa', 'Mars', To\_TIMESTAMP('2022-02-05 9:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.1, 'Round-Trip', 1, 'Fort Worth', 'Texas', 'United States', 'Earth', TO\_TIMESTAMP('2022-10-04 1:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '984230491','810091665', TO\_DATE('2021-07-02', 'YYYY-MM-DD'));**
2. **INSERT INTO TICKET VALUES('8192003', 'Spaceship', 'SpaceX', 60, 67, 'Nuwa', DEFAULT, 'Tempe Mensa', 'Mars', To\_TIMESTAMP('2022-02-05 9:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.1, 'Round-Trip', 1, 'Fort Worth', 'Texas', 'United States', 'Earth', TO\_TIMESTAMP('2022-10-04 1:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '721230456','810091665', TO\_DATE('2021-07-02', 'YYYY-MM-DD'));**
3. **INSERT INTO TICKET VALUES('8192004', 'Spaceship', 'SpaceX', 60, 67, 'Nuwa', DEFAULT, 'Tempe Mensa', 'Mars', To\_TIMESTAMP('2022-02-05 9:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.1, 'Round-Trip', 1, 'Fort Worth', 'Texas', 'United States', 'Earth', TO\_TIMESTAMP('2022-10-04 1:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '644578149','810091665', TO\_DATE('2021-07-02', 'YYYY-MM-DD'));**
4. **INSERT INTO TICKET VALUES('6651001', 'Train', 'Japan Railways', 65, 71, 'Tokyo', DEFAULT, 'Japan', 'Earth', To\_TIMESTAMP('2022-02-05 9:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.1, 'One-Way', 1, 'Osaka', 'DEFAULT', 'Japan', 'Earth', TO\_TIMESTAMP('2022-02-05 7:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '432779156','810091661', TO\_DATE('2022-02-05', 'YYYY-MM-DD'));**
5. **INSERT INTO TICKET VALUES('6651002', 'Train', 'Japan Railways', 65, 71, 'Tokyo', DEFAULT, 'Japan', 'Earth', To\_TIMESTAMP('2022-02-05 9:30:00', 'YYYY-MM-DD HH:MI:SS'), 0.1, 'One-Way', 1, 'Osaka', 'DEFAULT', 'Japan', 'Earth', TO\_TIMESTAMP('2022-02-05 7:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '432779157','810091661', TO\_DATE('2022-02-05', 'YYYY-MM-DD'));**
6. **INSERT INTO TICKET VALUES('6651003', 'Plane', 'Japan Airlines', 65, 71, 'Beirut', DEFAULT, 'Lebanon', 'Earth', To\_TIMESTAMP('2022-02-05 20:30:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', 1, 'Osaka', 'DEFAULT', 'Japan', 'Earth', TO\_TIMESTAMP('2022-02-05 12:30:00', 'YYYY-MM-DD HH:MI:SS'), '6651001', '432779156','810091662', TO\_DATE('2021-01-03', 'YYYY-MM-DD'));**
7. **INSERT INTO TICKET VALUES('6651004', 'Plane', 'Japan Airlines', 65, 71, 'Beirut', DEFAULT, 'Lebanon', 'Earth', To\_TIMESTAMP('2022-02-05 20:30:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', 1, 'Osaka', 'DEFAULT', 'Japan', 'Earth', TO\_TIMESTAMP('2022-02-05 12:30:00', 'YYYY-MM-DD HH:MI:SS'), '6651002', '432779157','810091662', TO\_DATE('2021-01-03', 'YYYY-MM-DD'));**
8. **INSERT INTO TICKET VALUES('7001920', 'Plane', 'Aeroitalia', 60, 78, 'Copenhagen', DEFAULT, 'Denmark', 'Earth', To\_TIMESTAMP('2022-05-11 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', 1, 'Milan', 'DEFAULT', 'Italy', 'Earth', TO\_TIMESTAMP('2022-05-11 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, '700192018','690192010', TO\_DATE('2022-04-03', 'YYYY-MM-DD'));**
9. **INSERT INTO TICKET VALUES('6701921', 'Plane', 'Middle East Airline', 60, 79, 'Paris', DEFAULT, 'France', 'Earth', To\_TIMESTAMP('2022-07-22 13:35:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', 1, 'Beirut', 'DEFAULT', 'Lebanon', 'Earth', TO\_TIMESTAMP('2022-07-22 09:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '556891162','187102918', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
10. **INSERT INTO TICKET VALUES('6701922', 'Plane', 'Middle East Airline', 60, 79, 'Paris', DEFAULT, 'France', 'Earth', To\_TIMESTAMP('2022-07-22 13:35:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', 1, 'Beirut', 'DEFAULT', 'Lebanon', 'Earth', TO\_TIMESTAMP('2022-07-22 09:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '556891162','187102918', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
11. **INSERT INTO TICKET VALUES('6701923', 'Plane', 'Middle East Airline', 60, 79, 'Paris', DEFAULT, 'France', 'Earth', To\_TIMESTAMP('2022-07-22 13:35:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', 1, 'Beirut', 'DEFAULT', 'Lebanon', 'Earth', TO\_TIMESTAMP('2022-07-22 09:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '556891162','187102918', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
12. **INSERT INTO TICKET VALUES('6701924', 'Plane', 'Middle East Airline', 60, 79, 'Paris', DEFAULT, 'France', 'Earth', To\_TIMESTAMP('2022-07-22 13:35:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', 1, 'Beirut', 'DEFAULT', 'Lebanon', 'Earth', TO\_TIMESTAMP('2022-07-22 09:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '556891162','187102918', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
13. **INSERT INTO TICKET VALUES('6701925', 'Plane', 'Middle East Airline', 60, 79, 'Paris', DEFAULT, 'France', 'Earth', To\_TIMESTAMP('2022-07-22 13:35:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '1', 'Beirut', 'DEFAULT', 'Lebanon', 'Earth', TO\_TIMESTAMP('2022-07-22 09:30:00', 'YYYY-MM-DD HH:MI:SS'), NULL, '556891162','187102918', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
14. **INSERT INTO TICKET VALUES('6701931', 'Plane', 'French Airlines', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '1', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701921', '556891162', '920267029', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
15. **INSERT INTO TICKET VALUES('6701932', 'Plane', 'French Airlines', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '1', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701922', '556891162','920267029', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
16. **INSERT INTO TICKET VALUES('6701933', 'Plane', 'French Airlines', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '1', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701923', '556891162','920267029', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
17. **INSERT INTO TICKET VALUES('6701934', 'Plane', 'French Airline', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '1', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701924', '556891162','920267029', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
18. **INSERT INTO TICKET VALUES('6701935', 'Plane', 'French Airline', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '1', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701925', '556891162','920267029', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
19. **INSERT INTO TICKET VALUES('6701941', 'Plane', 'American Airlines', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '0', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701921', NULL,'376180927', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
20. **INSERT INTO TICKET VALUES('6701942', 'Plane', 'American Airlines', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '0', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701922', NULL,'376180927', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
21. **INSERT INTO TICKET VALUES('6701943', 'Plane', 'American Airlines', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '0', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701923', NULL,'376180927', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
22. **INSERT INTO TICKET VALUES('6701944', 'Plane', 'American Airlines', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '0', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701924', NULL,'376180927', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
23. **INSERT INTO TICKET VALUES('6701945', 'Plane', 'American Airlines', 60, 52, 'Detroit', 'Michigan', 'United States', 'Earth', To\_TIMESTAMP('2022-07-23 15:25:00', 'YYYY-MM-DD HH24:MI:SS'), 0.1, 'One-Way', '0', 'Paris', 'DEFAULT', 'France', 'Earth', TO\_TIMESTAMP('2022-07-22 13:25:00', 'YYYY-MM-DD HH24:MI:SS'), '6701925', NULL,'376180927', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));**
    1. **HOTEL:**
24. **INSERT INTO HOTEL VALUES('754125479', '** **Liberty Motel', '43629', 'New York', 'New York', 'United States', 'Earth', '3','2','0.01','N');**
25. **INSERT INTO HOTEL VALUES('125227690', '** **Hogwarts School’, '30901', 'Hogwarts', NULL, 'Scotland', 'Earth', '5','3','0.02','N');**
26. **INSERT INTO HOTEL VALUES('620390011', 'King Galaxy Hotel', '00389', 'Nuwa', NULL, 'Tempe Mensa', 'Mars', '3','3','0.05','N');**
27. **INSERT INTO HOTEL VALUES('720901923', 'Four Seasons Hotel', '09135', 'Copenhagen', NULL, 'Denmark', 'Earth','4','3','0.05','N');**
28. **INSERT INTO HOTEL VALUES('720901924', 'Marriott Hotel', '09135', 'Copenhagen', NULL, 'Denmark', 'Earth', '4','3','0.05','N');**
29. **INSERT INTO HOTEL VALUES('720901925', 'Copenhagen Hotel', '09132', 'Copenhagen', NULL, 'Denmark', 'Earth', '2','2','0.02','N');**
30. **INSERT INTO HOTEL VALUES('720901926', 'Northern Light Hotel', '09131', 'Copenhagen', NULL, 'Denmark', 'Earth', '2','2','0.02','N');**
31. **INSERT INTO HOTEL VALUES('720901927', 'European Hotel', '09137', 'Copenhagen', NULL, 'Denmark', 'Earth', '4','2','0.05','N');**
32. **INSERT INTO HOTEL VALUES('720901928', 'Galaxy Hotel', '09138', 'Copenhagen', NULL, 'Denmark', 'Earth', '3','3','0.03','N');**
33. **INSERT INTO HOTEL VALUES('430293817', 'Hilton Garden Inn', '09135', 'Detroit', NULL, 'Michigan', 'Earth', '3','3','0.05','N');**
    1. **ROOM:** 
       1. **INSERT INTO ROOM VALUES('754125479', '122', '2', 'Double Bed');**
       2. **INSERT INTO ROOM VALUES('125227690', '871', '3', 'Triple Bed');**
       3. **INSERT INTO ROOM VALUES('620390011', '289', '4', 'Suite');**
       4. **INSERT INTO ROOM VALUES('720901923', '289', '1', 'King-Sized Bed');**
       5. **INSERT INTO ROOM VALUES('720901924', '288', '1', 'King-Sized Bed');**
       6. **INSERT INTO ROOM VALUES('720901924', '311', '1', 'Single Bed');**
       7. **INSERT INTO ROOM VALUES('720901924', '312', '1', 'Double Bed');**
       8. **INSERT INTO ROOM VALUES('720901924', '313', '1', 'Queen-Sized Bed');**
       9. **INSERT INTO ROOM VALUES('720901924', '314', '1', 'King-Sized Bed');**
       10. **INSERT INTO ROOM VALUES('430293817', '401', '5', 'Suite');**
    2. **CONTRACT**:
       1. **INSERT INTO CONTRACT VALUES('9876531', 'Y', 'Y', TO\_DATE('2022-01-10','YYYY-MM-DD'), 'Vacation', '** **889928456', '127362306');**
       2. **INSERT INTO CONTRACT VALUES('6170967', 'Y', 'Y', TO\_DATE('2022-01-10','YYYY-MM-DD'), 'School', '439088731', '781457987');**
       3. **INSERT INTO CONTRACT VALUES('6170922', 'Y', 'Y', TO\_DATE('2022-08-09', 'YYYY-MM-DD'),'Exploration', '** **439088731', '984230491');**
       4. **INSERT INTO CONTRACT VALUES('6170923', 'Y', 'Y', TO\_DATE('2022-08-09', 'YYYY-MM-DD'),'Exploration', '** **439088732', '721230456');**
       5. **INSERT INTO CONTRACT VALUES('6170924', 'Y', 'Y', TO\_DATE('2022-08-09', 'YYYY-MM-DD'), 'Exploration', '** **439088732', '644578149');**
       6. **INSERT INTO CONTRACT VALUES('6170933', 'Y', 'Y', TO\_DATE('2023-08-05', 'YYYY-MM-DD'), 'Music Festival', '** **819026609', '432779156');**
       7. **INSERT INTO CONTRACT VALUES('6170934', 'Y', 'Y', TO\_DATE('2023-08-05', 'YYYY-MM-DD'), 'Music Festival', '** **819026609', '432779157');**
       8. **INSERT INTO CONTRACT VALUES('6170959', 'Y', 'Y', TO\_DATE('2022-08-09', 'YYYY-MM-DD'), 'Business', '** **301928371', '700192018');**
       9. **INSERT INTO CONTRACT VALUES('6102918', 'N', 'N', TO\_DATE('2023-08-21', 'YYYY-MM-DD'), 'Business', '** **301928371', '700192018');**
       10. **INSERT INTO CONTRACT VALUES('3278001', 'Y', 'Y', TO\_DATE('2022-08-22', 'YYYY-MM-DD'), 'Immigration', '** **801927541', '556891162');**
    3. **COMPANION:**
34. **INSERT INTO COMPANION VALUES('Kevin', 'Josh', 'McCallister', '127362306', '13219784',** **TO\_DATE('2013-08-17','YYYY-MM-DD'), 'M', 'Son');**
35. **INSERT INTO COMPANION VALUES('Ron', 'Bilius', 'Weasley', '781457987', '81029767', TO\_DATE('1980-03-01','YYYY-MM-DD'), 'M', 'Friend');**
36. **INSERT INTO COMPANION VALUES('Hermione', 'Jean', 'Granger', '781457987', '81029768', TO\_DATE('1979-09-19', 'YYYY-MM-DD'), 'F', 'Friend');**
37. **INSERT INTO COMPANION VALUES('Ginny', 'Molly', 'Weasley', '781457987', '81029769', TO\_DATE('1982-09-25', 'YYYY-MM-DD'), 'F', 'Friend');**
38. **INSERT INTO COMPANION VALUES('Draco', 'Lucius', 'Malfoy', '781457987', '81029770', TO\_DATE('1979-03-12', 'YYYY-MM-DD'), 'M', 'Enemy');**
39. **INSERT INTO COMPANION VALUES('Severus', 'Alan', 'Snape', '781457987', '81029771', TO\_DATE('1955-04-17', 'YYYY-MM-DD'), 'M', 'Professor');**
40. **INSERT INTO COMPANION VALUES('Rima', 'Sana', 'Ammar', '556891162', '71820901', TO\_DATE('1998-06-12', 'YYYY-MM-DD'), 'F', 'Wife');**
41. **INSERT INTO COMPANION VALUES('Rana', 'Samira', 'Ammar', '556891162', '71820902', TO\_DATE('2018-07-12', 'YYYY-MM-DD'), 'F', 'Daughter');**
42. **INSERT INTO COMPANION VALUES('Rami', 'Hadi', 'Ammar', '556891162', '71820903', TO\_DATE('2019-06-12', 'YYYY-MM-DD'), 'M', 'Son');**
43. **INSERT INTO COMPANION VALUES('Rayan', 'Amir', 'Ammar', '556891162', '71820904', TO\_DATE('2020-06-22', 'YYYY-MM-DD'), 'M', 'Son');**
    1. **BILL:** 
       1. **INSERT INTO BILL VALUES('127918ABCDEF', TO\_DATE('2021-12-01', 'YYYY-MM-DD'), 'Crypto Wallet', '127362306');**
       2. **INSERT INTO BILL VALUES('18173FEDCBA', TO\_DATE('2021-09-04', 'YYYY-MM-DD'), 'Crypto Wallet', '781457987');**
       3. **INSERT INTO BILL VALUES('23456ABCDEF', TO\_DATE('2021-07-02', 'YYYY-MM-DD'), 'Crypto Wallet', '984230491');**
       4. **INSERT INTO BILL VALUES('23457ABCDEF ', TO\_DATE('2021-07-02', 'YYYY-MM-DD'), 'Crypto Wallet', '721230456');**
       5. **INSERT INTO BILL VALUES('23458ABCDEF ', TO\_DATE('2021-07-02', 'YYYY-MM-DD'), 'Crypto Wallet', '644578149');**
       6. **INSERT INTO BILL VALUES('56390ABCDEF ', TO\_DATE('2022-02-05', 'YYYY-MM-DD'), 'Crypto Wallet', '432779156');**
       7. **INSERT INTO BILL VALUES('56391ABCDEF ', TO\_DATE('2022-02-05', 'YYYY-MM-DD'), 'Crypto Wallet', '432779157');**
       8. **INSERT INTO BILL VALUES('91029ABCDEF ', TO\_DATE('2022-08-09', 'YYYY-MM-DD'), 'Crypto Wallet', '700192018');**
       9. **INSERT INTO BILL VALUES('91030ABCDEF ', TO\_DATE('2023-08-21', 'YYYY-MM-DD'), 'Crypto Wallet', '700192018');**
       10. **INSERT INTO BILL VALUES('46710ABCDEF ', TO\_DATE('2022-04-01', 'YYYY-MM-DD'), 'Crypto Wallet', '556891162');**
    2. **DEPARTMENT:** 
       1. **INSERT INTO DEPARTMENT VALUES(**'**00001', 'Research and Development', 'Responsible for developing the workspace and preventing company risks');**
       2. **INSERT INTO DEPARTMENT VALUES('00001', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       3. **INSERT INTO DEPARTMENT VALUES('00001', 'Customer Service', 'Responsible for attending to the customer needs');**
       4. **INSERT INTO DEPARTMENT VALUES('00001', 'Marketing Department', 'Responsible for promoting the business and attracting new customers');**
       5. **INSERT INTO DEPARTMENT VALUES('00001', 'Partner Relations', 'Responsible for maintaining and developing the relations with the partners of the company');**
       6. **INSERT INTO DEPARTMENT VALUES('00016', 'Research and Development', 'Responsible for developing the workspace and preventing company risks');**
       7. **INSERT INTO DEPARTMENT VALUES('00016', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       8. **INSERT INTO DEPARTMENT VALUES('00016', 'Customer Service', 'Responsible for attending to the customers needs');**
       9. **INSERT INTO DEPARTMENT VALUES('00016', 'Partner Relations', 'Responsible for maintaining and developing the relations with the partners of the company');**
       10. **INSERT INTO DEPARTMENT VALUES('00030', 'Research and Development', 'Responsible for developing the workspace and preventing company risks');**
       11. **INSERT INTO DEPARTMENT VALUES('00030', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       12. **INSERT INTO DEPARTMENT VALUES('00030', 'Customer Service', 'Responsible for attending to the customers needs');**
       13. **INSERT INTO DEPARTMENT VALUES('00030', 'Marketing Department', 'Responsible for promoting the business and attracting new customers');**
       14. **INSERT INTO DEPARTMENT VALUES('00030', 'Partner Relations', 'Responsible for maintaining and developing the relations with the partners of the company');**
       15. **INSERT INTO DEPARTMENT VALUES('00088', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       16. **INSERT INTO DEPARTMENT VALUES('00088', 'Customer Service', 'Responsible for attending to the customers needs');**
       17. **INSERT INTO DEPARTMENT VALUES('00055', 'Research and Development', 'Responsible for developing the workspace and preventing company risks');**
       18. **INSERT INTO DEPARTMENT VALUES('00055', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       19. **INSERT INTO DEPARTMENT VALUES('00055', 'Customer Service', 'Responsible for attending to the customers needs');**
       20. **INSERT INTO DEPARTMENT VALUES('00055', 'Marketing Department', 'Responsible for promoting the business and attracting new customers');**
       21. **INSERT INTO DEPARTMENT VALUES('00055', 'Partner Relations', 'Responsible for maintaining and developing the relations with the partners of the company');**
       22. **INSERT INTO DEPARTMENT VALUES('00066', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       23. **INSERT INTO DEPARTMENT VALUES('00066', 'Customer Service', 'Responsible for attending to the customers needs');**
       24. **INSERT INTO DEPARTMENT VALUES('00066', 'Partner Relations', 'Responsible for maintaining and developing the relations with the partners of the company');**
       25. **INSERT INTO DEPARTMENT VALUES('00077', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       26. **INSERT INTO DEPARTMENT VALUES('00077', 'Customer Service', 'Responsible for attending to the customers needs');**
       27. **INSERT INTO DEPARTMENT VALUES('00077', 'Partner Relations', 'Responsible for maintaining and developing the relations with the partners of the company');**
       28. **INSERT INTO DEPARTMENT VALUES('00044', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       29. **INSERT INTO DEPARTMENT VALUES('00044', 'Customer Service', 'Responsible for attending to the customers needs');**
       30. **INSERT INTO DEPARTMENT VALUES('00099', 'Customer Service', 'Responsible for promoting the business and attracting new customers');**
       31. **INSERT INTO DEPARTMENT VALUES('00099', 'Partner Relations', 'Responsible for maintaining and developing the relations with the partners of the company');**
       32. **INSERT INTO DEPARTMENT VALUES('00022', 'Human Resources', 'Responsible for recruiting, hiring, training, firing and overall managing the employees');**
       33. **INSERT INTO DEPARTMENT VALUES('00022', 'Customer Service', 'Responsible for attending to the customers needs');**
    3. **EMERGENCY\_CONTACT1:** 
       1. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Kate',** '**Megan', 'McCallister', '61728901, 'Wife', '127362306');**
       2. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Rubeus',** '**Albus', 'Hagrid', '90184110', 'Guardian', '781457987');**
       3. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Amy',** '**Waters', 'Davidson', '71912981', 'Mother', '984230491');**
       4. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Harry',** '**Sam', 'Holland', '71912982', 'Brother', '721230456');**
       5. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Blake',** '**Lively', 'Reynolds', '71912983', 'Brother', '644578149');**
       6. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Jane',** '**Alexia', 'Taylor', '71912984', 'Wife', '432779156');**
       7. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Alexandra',** '**Beatrice', 'Taylor', '71912985', 'Wife', '432779157');**
       8. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Mikhail',** '**Artyom', 'Lavrov', '71912986', 'Father', '601928374');**
       9. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Igor',** '**Konstantin', 'Pavlov', '71912987', 'Father', '601928375');**
       10. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Marco',** '**Gerard', 'Rossi', '78192019', 'Husband', '700192018');**
       11. **INSERT INTO EMERGENCY\_CONTACT1 VALUES(**'**Aanya',** '**Jiya', 'Patil', '78192901', 'Mother', '149823934');**
       12. **INSERT INTO EMERGENCY\_CONTACT1 VALUES('Raya', 'Zeina' ,'Ammar','56029274', '** **Mother', '556891162');**
    4. **EMERGENCY\_CONTACT2:**
       1. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Ava', 'Kayla', 'Maddison', '71817136', 'Wife', '889928456');**
       2. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Tobi', 'Greg', 'Smith', '71817129', '** **Brother', '61229982');**
       3. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Emma', 'Maddie', 'Williams', '91025567', '** **Wife', '439088731');**
       4. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Joshua', 'Tucker', 'Allen', '61720918', '** **Husband', '420885610');**
       5. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Mary', 'Miley', 'Coleman', '61720919', 'Sister', '439088732');**
       6. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Jack', 'Marc', 'Robert', '61720920', '** **Husband', '910284334');**
       7. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Kiyoko', 'Emiko', 'Kiyama', '71920912', '** **Brother', '819026609');**
       8. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Hari', 'Kin', 'Sakai', '71920913', '** **Mother', '819026608');**
       9. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Bako', 'Akin', 'Egbe', '80920914', '** **Father', '****611209182');**
       10. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Adewale', 'Dayo', 'Okoye', '80920915', '** **Brother', '611209181');**
       11. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Alina', 'Irina', 'Smirnov', '80920914', '** **Sister', '611209182');**
       12. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Viktor', 'Andrei', 'Orlov', '80920915', '** **Brother', '611209181');**
       13. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Giovanni', 'Lorenzo', 'Bruno', '80920916', '** **Brother', '301928371');**
       14. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Lia', 'Capri', 'Greco', '80920917', '** **Wife', '301928372');**
       15. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Diya', 'Prisha', 'Sharma', '80920918', '** **Wife', '****491022860');**
       16. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Riya', 'Myra', 'Singh', '80920919', '** **Wife', '491022861');**
       17. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Khalil', 'Samir', 'Saad', '51029888', '** **Husband', '801927541');**
       18. **INSERT INTO EMERGENCY\_CONTACT2 VALUES('Samer', 'Karim', 'Mrad', '51029889', '** **Husband', '801927542');**
    5. **BRANCH:** 
       1. **INSERT INTO BRANCH VALUES('00001', '00064', 'Chicago', 'Illinois', 'United States', 'Earth');**
       2. **INSERT INTO BRANCH VALUES('00016', '76809', 'London', NULL, 'United Kingdom', 'Earth');**
       3. **INSERT INTO BRANCH VALUES('00030', '71928', 'Fort Worth', 'Texas', 'United States', 'Earth');**
       4. **INSERT INTO BRANCH VALUES('00088', '00251', 'Osaka', NULL, 'Japan', 'Earth');**
       5. **INSERT INTO BRANCH VALUES('00055', '00809', 'Lagos', NULL, 'Nigeria', 'Earth');**
       6. **INSERT INTO BRANCH VALUES('00066', '61928', 'Moscow', NULL, 'Russia', 'Earth');**
       7. **INSERT INTO BRANCH VALUES('00077', '11029', 'Astana', NULL, 'Kazakhstan', 'Earth');**
       8. **INSERT INTO BRANCH VALUES('00044', '65901', 'Milan', NULL, 'Italy', 'Earth');**
       9. **INSERT INTO BRANCH VALUES('00099', '28102', 'New Delhi', NULL, 'India', 'Earth');**
       10. **INSERT INTO BRANCH VALUES('00022', '91029', ‘Beirut’, NULL, 'Lebanon', 'Earth');**
    6. **PARTNER:** 
       1. **INSERT INTO PARTNER VALUES('376180927', 'American Airlines', 'Fort Worth', 'Texas', 'United States', 'Earth', 'Flights', '56198');**
       2. **INSERT INTO PARTNER VALUES('541180927', 'Hogwarts Express', 'London', NULL, 'England', 'Earth', 'Trains', '37116');**
       3. **INSERT INTO PARTNER VALUES('810091665', 'SpaceX', 'Fort Worth', 'Texas', 'United States', 'Earth', 'Spaceships', '37133');**
       4. **INSERT INTO PARTNER VALUES('810091661', 'Japan Railways', 'Osaka', NULL, 'Japan', 'Earth', 'Trains', '57109');**
       5. **INSERT INTO PARTNER VALUES('810091662', 'Japan Airline', 'Osaka', NULL, 'Japan', 'Earth', 'Airplanes', '16273');**
       6. **INSERT INTO PARTNER VALUES('779018291', 'Russia Airline', 'Moscow', NULL, 'Russia', 'Earth', 'Airplanes', '23789');**
       7. **INSERT INTO PARTNER VALUES('779018292', 'Russia Railways', 'Kazan', NULL, 'Russia', 'Earth', 'Trains', '30192');**
       8. **INSERT INTO PARTNER VALUES('779018293', 'Russia Busses', 'St. Petersburg', NULL, 'Russia', 'Earth', 'Trains', '30192');**
       9. **INSERT INTO PARTNER VALUES('690192010', 'Aeroitalia', 'Rome', NULL, 'Italy', 'Earth', 'Planes', '91029');**
       10. **INSERT INTO PARTNER VALUES('187102918', 'Middle East Airline', 'Beirut', NULL, 'Lebanon', 'Earth', 'Planes', '55020');**
       11. **INSERT INTO PARTNER VALUES('920267029', 'French Airlines', 'Paris', NULL, 'France', 'Earth', 'Planes', '00020');**
    7. **CUSTOMER\_PURCHASE\_HISTORY:**
       1. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('1457855', '127362306');**
       2. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('1588511', '781457987');**
       3. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('8192002', '984230491');**
       4. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('8192003', '721230456');**
       5. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('8192004', '644578149');**
       6. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('6651001', '432779156');**
       7. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('6651002', '432779157');**
       8. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('6651003', '432779156');**
       9. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('6651004', '432779157');**
       10. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('7001920', '700192018');**
       11. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('6701921', '556891162');**
       12. **INSERT INTO CUSTOMER\_PURCHASE\_HISTORY VALUES('6701922', '556891162');**
    8. **PARTNER\_OPERATING\_COUNTRIES:** 
       1. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('United States', '376180927');**
       2. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('England', '5411809272');**
       3. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Scotland', '5411809272');**
       4. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Wales', '5411809272');**
       5. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Northern Ireland', '5411809272');**
       6. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('United States', '810091665');**
       7. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Japan', '810091661');**
       8. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Japan', '810091661');**
       9. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Russia', '779018291');**
       10. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Russia', '779018292');**
       11. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Russia','779018293');**
       12. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Italy', '690192010');**
       13. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('Lebanon', '187102918');**
       14. **INSERT INTO PARTNER\_OPERATING\_COUNTRIES VALUES('France', '920267029');**
    9. **RECOMMENDS:** 
       1. **INSERT INTO RECOMMENDS VALUES('889928456', '127362306', '754125479');**
       2. **INSERT INTO RECOMMENDS VALUES('439088731', '781457987', '125227690');**
       3. **INSERT INTO RECOMMENDS VALUES('439088732', '984230491', '620390011');**
       4. **INSERT INTO RECOMMENDS VALUES('439088732', '721230456', '620390011');**
       5. **INSERT INTO RECOMMENDS VALUES('439088732', '644578149', '620390011');**
       6. **INSERT INTO RECOMMENDS VALUES('301928371', '700192018', '720901923');**
       7. **INSERT INTO RECOMMENDS VALUES('301928371', '700192018', '720901925');**
       8. **INSERT INTO RECOMMENDS VALUES('301928371', '700192018', '720901926');**
       9. **INSERT INTO RECOMMENDS VALUES('301928371', '700192018', '720901927');**
       10. **INSERT INTO RECOMMENDS VALUES('801927541', '556891162', '430293817');**
    10. **MANAGES:** 
        1. **INSERT INTO MANAGES VALUES('889928456', '127362306');**
        2. **INSERT INTO MANAGES VALUES('439088731', '781457987');**
        3. **INSERT INTO MANAGES VALUES('439088732', '984230491');**
        4. **INSERT INTO MANAGES VALUES('439088732', '721230456');**
        5. **INSERT INTO MANAGES VALUES('439088732', '644578149');**
        6. **INSERT INTO MANAGES VALUES('819026609', '432779156');**
        7. **INSERT INTO MANAGES VALUES('819026609', '432779157');**
        8. **INSERT INTO MANAGES VALUES('344309012', '601928374');**
        9. **INSERT INTO MANAGES VALUES('344309012', '601928375');**
        10. **INSERT INTO MANAGES VALUES('301928371', '700192018');**
        11. **INSERT INTO MANAGES VALUES('491022860', '149823934');**
        12. **INSERT INTO MANAGES VALUES('801927541', '556891162');**
44. **FINAL TABLES STATE:**
    1. **CUSTOMER**



* 1. **EMPLOYEE**





* 1. **TICKET**







* 1. **HOTEL**
  2. **ROOM**



* 1. **CONTRACT**
  2. **COMPANION**



* 1. **BILL**



* 1. **DEPARTMENT**
  2. **EMERGENCY\_CONTACT1**
  3. **EMERGENCY\_CONTACT2**
  4. **BRANCH**



* 1. **PARTNER**
  2. **CUSTOMER\_PURCHASE\_HISTORY**



* 1. **PARTNER\_OPERATING\_COUNTRIES**



* 1. **RECOMMENDS**



* 1. **MANAGES**



1. **Queries:**
   1. **Alone in New York:**

On Christmas eve, Kevin McCallister, a lost 10-year-old boy at the airport, bumped into a passenger check-in officer at the virtual Chicago O'Hare International Airport, causing her to lose her balance and scatter the tickets in her hand. As the plane heading to New York City was about to departure, she allowed Kevin to board the plane without searching for and checking his ticket. However, before takeoff and after reorganizing the tickets, she found that there was a distinct ticket for a plane headed to New York City, New York. She contacted the airport security who, in turn, contacted Metatravel agency in hopes of reaching Kevin’s father.

First, we need to confirm if Kevin is the customer by checking if the **Passport\_ID= 127362306** found on the ticket belongs to him. If yes, we need to inform Kevin that he is on the wrong flight by finding his **Chat\_ID.** Otherwise, the **Passport\_ID** belongs to a certain customer, whose companion is Kevin. In this case, we need to urgently identify that customer and find their **Chat\_ID in** order to contact them.

* **Code**

SELECT CUSTOMER.FIRST\_NAME, CUSTOMER.MIDDLE\_NAME, CUSTOMER.LAST\_NAME, CUSTOMER.Chat\_id

FROM CUSTOMER

WHERE CUSTOMER.Passport\_id = '784725619'

UNION

SELECT CUSTOMER.FIRST\_NAME, CUSTOMER.MIDDLE\_NAME, CUSTOMER.LAST\_NAME, CUSTOMER.Chat\_id

FROM CUSTOMER, COMPANION

WHERE CUSTOMER.Passport\_id=COMPANION.P\_id AND COMPANION.First\_Name='Kevin' AND COMPANION.Middle\_Name = 'Josh' AND COMPANION.Last\_Name='McCallister';

* **Result**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Result\_Entry** | **FIRST\_NAME** | **MIDDLE\_NAME** | **LAST\_NAME** | **CHAT\_ID** |
| **1** | Peter | Austin | McCallister | 85764399 |

* 1. **Glitch at Hogwarts Express: Mischief Managed**

Harry Potter, along with his companions Ron Weasley, Hermione Granger, Ginny Weasley, Draco Malfoy, and Severus Snape, are headed towards Hogwarts School of Witchcraft and Wizardry to defeat Lord Voldemort. A glitch occurred in their train right before departure. The train was heading from **City = “London”, Country= “England”, Planet = “Earth”** on **'2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'** to **City = “Hogwarts,” Country= “Scotland”, Planet = “Earth”** arriving at **'2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS'**. To avoid further delays, we want to provide our customers with a list of available train tickets that are heading to this location at this time.

* **Code:**

SELECT Ticket\_number

FROM TICKET

WHERE Departure\_City = 'London' AND Departure\_Country = 'England' AND Destination\_City = 'Hogwarts' AND Destination\_City = 'Hogwarts' AND Destination\_Country = 'Scotland' AND Departure\_Time = To\_TIMESTAMP('2021-10-04 11:30:00', 'YYYY-MM-DD HH:MI:SS') AND Destination\_Time = To\_TIMESTAMP('2021-10-04 12:30:00', 'YYYY-MM-DD HH:MI:SS') AND Psprt\_id IS NULL;

* **Result**



1. **Fall of Bitcoin:**

As part of a new project, the Research and Development Department at **Branch number 00055** has employed a machine learning model to predict the fluctuations of Bitcoin prices. Bitcoin price values are projected to decrease by 45% in two months. To avoid bankruptcy, Metatravel has decided to convert the currency it mainly uses from Bitcoin to a more secure currency - Ethereum. Therefore, the salaries of the employees and the prices of its services will change to the cryptocurrency Ethereum.

Formula to convert from BTC to ETH: x= (x\*13.74)

After of a couple of months, Bitcoin’s value did indeed crash. The predictions of Research and Development Department at that specific branch were accurate, so Metatravel has decided to award all the employees working for the branch a 10% raise to their salaries.

* **Code:**
  + SELECT HOTEL\_ID, HOTEL\_NAME, PRICE

FROM HOTEL;

* + SELECT TICKET\_NUMBER, PRICE

FROM TICKET;

* + SELECT EMPLOYEE\_SSN, FIRST\_NAME, LAST\_NAME, SALARY

FROM EMPLOYEE;

* + UPDATE EMPLOYEE

SET Salary = Salary + (Salary \* 10 / 100)

WHERE BRANCH\_NUM = '00055';

* + UPDATE EMPLOYEE

SET Salary = Salary \* 13.74;

* + UPDATE HOTEL

SET PRICE = PRICE \* 13.74;

* + UPDATE TICKET

SET PRICE = PRICE \* 13.74;

* + SELECT HOTEL\_ID, HOTEL\_NAME, PRICE

FROM HOTEL;

* + SELECT TICKET\_NUMBER, PRICE

FROM TICKET;

* + SELECT EMPLOYEE\_SSN, FIRST\_NAME, LAST\_NAME, SALARY

FROM EMPLOYEE;

* **Result:**
  + **Before:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Result\_Number** | **EMPLOYEE\_SSN** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** |
| **1** | 61229982 | John | Smith | 7.02 |
| **2** | 889928456 | James | Maddison | 6.02 |
| **3** | 420885610 | Faith | Allen | 7.02 |
| **4** | 439088731 | Simon | Williams | 6.02 |
| **5** | 910284334 | Grace | Robert | 7.02 |
| **6** | 439088732 | Elizabeth | Coleman | 6.02 |
| **7** | 819026608 | Izumi | Sakai | 7.02 |
| **8** | 819026609 | Akira | Kiyama | 6.02 |
| **9** | 611209181 | Chinara | Okoye | 9.02 |
| **10** | 611209182 | Blessing | Egbe | 8.02 |
| **11** | 344309012 | Natasha | Smirnov | 5.02 |
| **12** | 344309013 | Ivan | Orlov | 7.02 |
| **13** | 301928372 | Alice | Greco | 7.02 |
| **14** | 301928371 | Eduardo | Bruno | 5.42 |
| **15** | 491022861 | Kiaan | Singh | 7.02 |
| **16** | 491022860 | Viraj | Sharma | 6.02 |
| **17** | 801927542 | Sarah | Mrad | 7.02 |
| **18** | 801927541 | Ghada | Saad | 6.02 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Result\_Number** | **HOTEL\_ID** | **HOTEL\_NAME** | **PRICE** |
| **1** | 754125479 | Liberty Motel | 0.01 |
| **2** | 620390011 | King Galaxy Hotel | 0.05 |
| **3** | 720901923 | Four Seasons Hotel | 0.05 |
| **4** | 720901924 | Marriott Hotel | 0.05 |
| **5** | 720901925 | Copenhagen Hotel | 0.02 |
| **6** | 720901926 | Northern Light Hotel | 0.02 |
| **7** | 720901927 | European Hotel | 0.05 |
| **8** | 720901928 | Galaxy Hotel | 0.03 |
| **9** | 430293817 | Hilton Garden Inn | 0.05 |
| **10** | 125227690 | Hogwarts School | 0.02 |



* + **After:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Result\_Number | EMPLOYEE\_SSN | FIRST\_NAME | LAST\_NAME | SALARY |
| 1 | 61229982 | John | Smith | 96.45 |
| 2 | 889928456 | James | Maddison | 82.71 |
| 3 | 420885610 | Faith | Allen | 96.45 |
| 4 | 439088731 | Simon | Williams | 82.71 |
| 5 | 910284334 | Grace | Robert | 96.45 |
| 6 | 439088732 | Elizabeth | Coleman | 82.71 |
| 7 | 819026608 | Izumi | Sakai | 96.45 |
| 8 | 819026609 | Akira | Kiyama | 82.71 |
| 9 | 611209181 | Chinara | Okoye | 136.3 |
| 10 | 611209182 | Blessing | Egbe | 121.19 |
| 11 | 344309012 | Natasha | Smirnov | 68.97 |
| 12 | 344309013 | Ivan | Orlov | 96.45 |
| 13 | 301928372 | Alice | Greco | 96.45 |
| 14 | 301928371 | Eduardo | Bruno | 74.47 |
| 15 | 491022861 | Kiaan | Singh | 96.45 |
| 16 | 491022860 | Viraj | Sharma | 82.71 |
| 17 | 801927542 | Sarah | Mrad | 96.45 |
| 18 | 801927541 | Ghada | Saad | 82.71 |



1. **Russian-Ukraine War Repercussion:**

After Vladimir Putin, Russia’s president, ordered the invasion of Ukraine on February 24, 2022, Metatravel, along with other companies, were faced with the dilemma of re-examining their ties with Russia. Despite the strong ties built with Russia during the last couple of years, Metatravel has decided to withdraw from Russia and relocate its Russian branch to Kazakhstan. Therefore, all employees working for the Russian branch with **Branch\_Number=00066** have to be transferred to the Kazakh branch with **Branch\_Number=00077**.

Moreover, Metatravel has decided to cut ties with all of its partners still operating in Russia. Thus, partners with **Country= “Russia”** should be deleted.

* **Code:**
  + SELECT EMPLOYEE\_SSN, FIRST\_NAME, LAST\_NAME, BRANCH\_NUM, BRANCH.COUNTRY

FROM EMPLOYEE, BRANCH

WHERE BRANCH\_NUM = '00066' AND BRANCH\_NUMBER = BRANCH\_NUM;

* + SELECT PARTNER\_ID, NAME, COUNTRY

FROM PARTNER

* + UPDATE EMPLOYEE

SET BRANCH\_NUM = '00077'

WHERE BRANCH\_NUM = '00066';

* + DELETE PARTNER

WHERE COUNTRY = 'Russia';

* + SELECT EMPLOYEE\_SSN, FIRST\_NAME, LAST\_NAME, BRANCH\_NUM, BRANCH.COUNTRY

FROM EMPLOYEE, BRANCH

WHERE BRANCH\_NUM = '00077' AND BRANCH\_NUMBER = BRANCH\_NUM;

* + SELECT PARTNER\_ID, NAME, COUNTRY

FROM PARTNER

* **Result:**
  + **Before:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Result\_Number** | **EMPLOYEE\_SSN** | **FIRST\_NAME** | **LAST\_NAME** | **BRANCH\_NUM** | **COUNTRY** |
| **1** | 344309012 | Natasha | Smirnov | 66 | Russia |
| **2** | 344309013 | Ivan | Orlov | 66 | Russia |



* + **After:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Result\_Number** | **EMPLOYEE\_SSN** | **FIRST\_NAME** | **LAST\_NAME** | **BRANCH\_NUM** | **COUNTRY** |
| **1** | 344309012 | Natasha | Smirnov | 77 | Kazakhstan |
| **2** | 344309013 | Ivan | Orlov | 77 | Kazakhstan |



1. **Roomergency:**

A travel agent at Metatravel recommended the wrong hotel to the customer **Andrea Sofia Rossi**. When Andrea arrived at the Four Seasons Hotel (**Hotel\_ID= 720901923**) in Denmark and tried to check-in, the hotel staff mentioned that there were no reservations under her name. Shocked, Andrea immediately contacted Metatravel. It turns out that the travel agent gave her a recommendation that did not match her expectations and wrongfully booked it for her.

Metatravel now has to rectify the situation and find a new hotel in the same area of **City= Copenhagen** and **Country= Denmark** with available rooms to book. Also, the agency apologized on behalf of the employee and decided to grant Andrea a 10% bonus to her points balance.

Additionally, Metatravel has to find the agent that made the wrong recommendation and decrease 10% from their salary. The employee will remain under probation until further notice.

Given:

- **Andrea’s Passport\_ID = 700192018**

-**Wrong Hotel\_ID= 720901923** (We need to find the room type requested by Andrea and the location of the hotel to match them to a new hotel)

* **Code:**
  + SELECT DISTINCT EMPLOYEE.EMPLOYEE\_SSN, EMPLOYEE.FIRST\_NAME, EMPLOYEE.LAST\_NAME, EMPLOYEE.SALARY

FROM EMPLOYEE, RECOMMENDS

WHERE EMPLOYEE.EMPLOYEE\_SSN = RECOMMENDS.EMPLOYEE\_SSN AND RECOMMENDS.Psprt\_id= '700192018';

* + SELECT FIRST\_NAME, LAST\_NAME, PASSPORT\_ID , POINTS\_BALANCE, HOTEL\_ID

FROM CUSTOMER

WHERE PASSPORT\_ID = '700192018';

* + UPDATE EMPLOYEE

SET Salary = Salary - (Salary \* 10 / 100)

WHERE EMPLOYEE.EPLOYEE\_SSN IN

(SELECT DISTINCT EMPLOYEE.EMPLOYEE\_SSN, EMPLOYEE.FIRST\_NAME, EMPLOYEE.LAST\_NAME, EMPLOYEE.SALARY

FROM EMPLOYEE, RECOMMENDS

WHERE EMPLOYEE.EMPLOYEE\_SSN = RECOMMENDS.EMPLOYEE\_SSN AND RECOMMENDS.Psprt\_id= '700192018');

* + UPDATE CUSTOMER

SET CUSTOMER.HOTEL\_ID =

(SELECT ROOM.H\_ID

FROM HOTEL, ROOM

WHERE HOTEL.HOTEL\_ID = ROOM.H\_ID AND HOTEL.CITY = 'Copenhagen' AND HOTEL.COUNTRY = 'Denmark' AND (NOT CUSTOMER.HOTEL\_ID = HOTEL.HOTEL\_ID) AND ROWNUM = 1)

WHERE CUSTOMER.Passport\_id IN

(SELECT DISTINCT CUSTOMER.Passport\_id

FROM CUSTOMER

WHERE CUSTOMER.PASSPORT\_ID = '700192018');

* + SELECT DISTINCT EMPLOYEE.EMPLOYEE\_SSN, EMPLOYEE.FIRST\_NAME, EMPLOYEE.LAST\_NAME, EMPLOYEE.SALARY

FROM EMPLOYEE, RECOMMENDS

WHERE EMPLOYEE.EMPLOYEE\_SSN = RECOMMENDS.EMPLOYEE\_SSN AND RECOMMENDS.Psprt\_id= '700192018';

* + SELECT FIRST\_NAME, LAST\_NAME, PASSPORT\_ID , POINTS\_BALANCE, HOTEL\_ID

FROM CUSTOMER

WHERE PASSPORT\_ID = '700192018';

* **Result:**
  + **Before:**





* + **After:**





1. **The Disposal of Expired Contracts:**

Metatravel has decided to dispose of all the expired contracts in its database to save up on storage space. Therefore, we need to create a query that deletes all the contracts from the contract table with expiry date before the current date.

* **Code:**
  + SELECT \* FROM CONTRACT;
  + DELETE FROM CONTRACT WHERE Expiry\_date < trunc(sysdate);
  + SELECT \* FROM CONTRACT;
* **Result:**
  + **Before:**



* + **After:**



1. **For the Love of Football:**

**Kumar Aarav Patil,** a Metatravel customer, is an avid football supporter. With the 2022 World Cup coming this November, Kumar was adamant that he was traveling to Qatar to watch it. His points balance with Metatravel is 92 points. He needed to meet the threshold of 100 to get a 50% discount on his flight. Not thinking clearly, Kumar, with the help of the Metatravel employee managing him, decided to hack into Metatravel’s database system and increase his balance to 100 in hopes of receiving a 50% discount. The database administrators at Metatravel were immediately notified of this breach. In this query, we seek to first find the employee who helped Kumar hack into the system and fire them. Moreover, we need to delete Kumar from the database since he has been banned from Metatravel. The dispute will later be settled in court.

Given:

**Kumar’s Passport\_ID= 432779157**

* **Code:**
  + SELECT EMPLOYEE\_SSN, FIRST\_NAME, LAST\_NAME FROM EMPLOYEE;
  + SELECT EMPLOYEE.EMPLOYEE\_SSN, EMPLOYEE.FIRST\_NAME, EMPLOYEE.LAST\_NAME

FROM EMPLOYEE, MANAGES

WHERE EMPLOYEE.EMPLOYEE\_SSN = MANAGES.EMPLOYEE\_SSN AND MANAGES.PSPRT\_ID = '432779157';

* + DELETE FROM EMPLOYEE

WHERE EXIST

(SELECT EMPLOYEE.EMPLOYEE\_SSN

FROM EMPLOYEE, MANAGES

WHERE EMPLOYEE.EMPLOYEE\_SSN = MANAGES.EMPLOYEE\_SSN AND MANAGES.PSPRT\_ID = '432779157');

* + DELETE FROM CUSTOMER WHERE PASSPORT\_ID = '**432779157';**
  + SELECT EMPLOYEE\_SSN, FIRST\_NAME, LAST\_NAME FROM EMPLOYEE;
* **Result:**
  + **Before:**





* + **AFTER:**



1. **Hijacked While Ad Astra[[2]](#footnote-2)**

With a vision of colonizing and establishing a self-sustaining city on Mars, SpaceX CEO Elon Musk has decided to organize a voyage to the Red Planet. However, the spaceship has been hijacked by a bitter Twitter employee, whom Musk has recently fired after taking control of Twitter in a $44 billion deal. The spacecraft pilot immediately reported to the mission control room at SpaceX’s headquarters in Texas that the spacecraft has been hijacked. However, it was too late.

The spacecraft soon exploded in space. 4 passengers were lost that day. One of them was Elon Musk, the world’s richest tech billionaire. Metatravel, who had arranged the trip for the other 3 passengers, was immediately informed of the tragedy. It is thus our duty to identify the 3 customers who were on that spacecraft and reach the emergency contacts to break the news to them. We need to find the **Names** and **Chat\_IDs** of the emergency contacts of each of the dead passengers knowing that their ticket numbers are **8192002, 8192003,** and **8192004.**

* **Code:**

SELECT DISTINCT EMERGENCY\_CONTACT1.First\_Name, EMERGENCY\_CONTACT1.Middle\_Name, EMERGENCY\_CONTACT1.Last\_Name, EMERGENCY\_CONTACT1.Chat\_ID, EMERGENCY\_CONTACT1.P\_ID

FROM EMERGENCY\_CONTACT1, CUSTOMER, TICKET

WHERE CUSTOMER.Passport\_ID = EMERGENCY\_CONTACT1.P\_ID AND CUSTOMER.PASSPORT\_ID IN

(SELECT TICKET.Psprt\_ID

FROM TICKET

WHERE Ticket\_number= '8192002' OR Ticket\_number= '8192003' OR Ticket\_number='8192004');

* **Result:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Result\_Number** | **FIRST\_NAME** | **MIDDLE\_NAME** | **LAST\_NAME** | **CHAT\_ID** | **P\_ID** |
| **1** | Harry | Sam | Holland | 71912982 | 721230456 |
| **2** | Amy | Waters | Davidson | 71912981 | 984230491 |
| **3** | Blake | Lively | Reynolds | 71912983 | 644578149 |

1. **Carlos Ghosn: Capturing the Fugitive**

Carlos Ghosn – the former CEO of the Renault-Nissan Alliance – has been charged with financial crimes in Japan. He had been previously arrested a few times but granted requests for bail. Ghosn claimed he “no longer be held hostage by a rigged Japanese justice system where guilt is presumed, discrimination is rampant, and basic human rights are denied.” On the 5th of February 2022, numerous media outlets reported Ghosn’s plan to escape Japan. He was seen outside of a Metatravel branch in Osaka with two men carrying large containers and a music equipment box in which Ghosn was allegedly supposed to hide.

The Interpol has issued a red notice for his arrest and has asked Metatravel to help find the fugitive. All that has been known is that the men are headed to a certain airport in Japan. However, they have used another mode of transportation before that. We need to find all the tickets that connect to a ticket with **Departure\_Country= “Japan”** and **Mode\_Of\_Transportation= “Plane.”** The tickets belong to two customers with **Gender= “M”** and **“Brnch\_nb= 00088”** We identify all the **Passport\_IDs** of these customers to ban them from travelling until the culprits are caught.

* **Code:**

SELECT Distinct CUSTOMER.Passport\_ID

FROM CUSTOMER, TICKET

WHERE CUSTOMER.Gender= 'M' AND CUSTOMER.Brnch\_nb= '00088' AND

TICKET.Ticket\_Number IN

(SELECT TICKET.Connection\_Number

FROM TICKET

WHERE Departure\_Country= 'Japan' AND Mode\_Of\_Transportation= 'Plane');

* **Result:**



1. **Arabophobia in the Air**

**Ibrahim Ammar** is a Lebanese customer of Metatravel who is in the process of immigrating to the United States. He recently booked tickets for him, his wife, and their 3 children to travel to the United States. However, due to the American ban on Middle East Airlines[[3]](#footnote-3), he had to book two connecting tickets: from Lebanon to France and from France to Detroit, Michigan. After the first flight and upon their arrival to Paris, Ibrahim and his family were racially profiled by French Airlines security. They were not allowed to board their connecting flight from France to Detroit with French Airlines.

Being unable to stay in France due to their lack of a Schengen visa, the Ammar family must immediately leave the country. Ibrahim contacts his Metatravel agent, explains the situation, and requests to book another flight. The new flights are booked with American Airlines. Appalled by the situation, the agent informs his supervisor, who in turn informs the Partner-Relations department at the Lebanese branch. Metatravel does not tolerate discrimination and decides to cut ties with French Airlines until an apology statement is issued.

Given:

**-Ibrahims’s passport\_id:** **81735301**

Steps:

-Find Ibrahim’s travel agent (through MANAGES relationship) + the employee’s supervisor ESSN

-Delete French Airlines from PARTNER table

-Find Ibrahim and his family members new tickets to travel to Detroit.

* **Code:**
  + 1) SELECT FIRST\_NAME, LAST\_NAME, EMPLOYEE\_SSN

FROM EMPLOYEE

WHERE EMPLOYEE.EMPLOYEE\_SSN IN

(SELECT EMPLOYEE.Supervisor\_SSN

FROM EMPLOYEE, MANAGES

WHERE EMPLOYEE.Employee\_SSN = MANAGES.Employee\_SSN AND MANAGES.Employee\_SSN IN

(SELECT MANAGES.Employee\_SSN

FROM MANAGES, CUSTOMER

WHERE CUSTOMER.Passport\_ID= '556891162' AND CUSTOMER.Passport\_ID= MANAGES.Psprt\_ID));

* + 2)SELECT \* FROM PARTNER;
  + DELETE PARTNER

WHERE NAME = 'French Airlines';

* + SELECT \* FROM PARTNER;
  + 3) SELECT TICKET\_NUMBER, TRANSPORT\_COMPANY, DESTINATION\_CITY, DEPARTURE\_CITY, CONNECTION\_NUMBER, PSPRT\_ID, PRTNR\_ID

FROM TICKET

WHERE PSPRT\_ID = '556891162';

* + UPDATE TICKET

SET PSPRT\_ID = '556891162'

WHERE TICKET\_NUMBER IN

(SELECT DISTINCT TICKET\_NUMBER

FROM TICKET

WHERE DESTINATION\_CITY = 'Detroit' AND DEPARTURE\_CITY = 'Paris');

* + SELECT TICKET\_NUMBER, TRANSPORT\_COMPANY, DESTINATION\_CITY, DEPARTURE\_CITY, CONNECTION\_NUMBER, PSPRT\_ID, PRTNR\_ID

FROM TICKET

WHERE PSPRT\_ID = '556891162';

**Result:**

1)

|  |  |  |  |
| --- | --- | --- | --- |
| **Result\_Number** | **FIRST\_NAME** | **LAST\_NAME** | **EMPLOYEE\_SSN** |
| **1** | Sarah | Mrad | 801927542 |

2)

Before:



After:



3)Before:



After:

1. **Normalization Up to BCNF**

Ensuing the creation of all relations, we should improve them by normalizing them according to several normal forms. Here, we will normalize them up to the Boyce-Codd Normal Form. In other words, we will normalize them according to the first normal form, followed by the second, third and Boyce-Codd normal forms.

**First Normal Form:**

This form dictates that each field within a relation schema should not be multivalued, composite, or their combinations. Only single atomic values are permitted as attribute values. Domain of an attribute must solely include atomic values, and the value of an attribute in a tuple must be a single value from the domain of that attribute. Namely, a set of values cannot be an attribute value for a single tuple.

**Second Normal Form**

To meet this form, all the nonprime attributes in the relation ought to be fully functionally dependent on the prime attributes, such that:

* A prime attribute is an attribute that is a member of the primary key K in a relation R. Hence, a nonprime attribute is not a member of any candidate key.
* A functional dependency X →Y is a constraint between two sets of attributes from the database. If the values of component Y of a tuple in relation R depend on or are determined by the values of an X component. We say that Y is functionally dependent on X.
* A functional dependency X →Y is a full functional dependency if the removal of any attribute A from X means that the functional dependency does not hold anymore. However, a functional dependency X →Y is a partial functional dependency if removal of any attribute A from X means that the dependency still holds.

**Third Normal Form**

A relation is in third normal form (3NF) if whenever a functional dependency X →Y holds, then either:

1. X is a superkey of R, or
2. Y is a prime attribute of R

**Boyce-Codd Normal Form**

A relation schema R is in Boyce-Codd Normal Form (BCNF) if whenever a functional dependency X→Y holds then X is a superkey of R. This normal form is a subset of the 3NF.

1. **CUSTOMER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Passport\_id | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth |

**FD1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Street\_Number | City | State | Country | Planet | Gender | Points\_balance | Stay\_Duration | Hotel\_id | Brnch\_nb |

1. The **CUSTOMER** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **CUSTOMER** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key Passport\_id.
3. The **CUSTOMER** relation schema satisfies the 3NF because Passport\_id is the superkey of that table (at least one of the conditions is satisfied).
4. The **CUSTOMER** relation schema satisfies the BCNF because Passport\_id is the superkey of that table.
5. **EMPLOYEE**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Essn | First\_Name | Middle\_Name | Last\_Name | Chat\_id | Date\_of\_Birth | Street\_Number | City | State |

**FD1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Position | Planet | Gender | Salary | Country | Supervisor\_ssn | Branch\_nb | Dept\_name |

1. The **EMPLOYEE** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **EMPLOYEE** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key Essn.
3. The **EMPLOYEE** relation schema satisfies the 3NF because Essn is the superkey of that table (at least one of the conditions is satisfied).
4. The **EMPLOYEE** relation schema satisfies the BCNF because Essn is the superkey of that table.
5. **TICKET**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Departure\_Time | Departure\_Date | Departure\_City | Departure\_State | Departure\_Country | Departure\_Planet | Price | Ticket\_type | Is\_booked |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ticket\_number | Mode\_of\_Transportation | Transport\_company | Baggage\_weight | Seat\_number | Destination\_Time | Destination\_Date |

**FD1**

**FD2**

|  |  |  |  |
| --- | --- | --- | --- |
| Destination\_City | Destination\_State | Destination\_Country | Destination\_Planet |

1. The **TICKET** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **TICKET** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key Ticket\_number.
3. The **TICKET** relation schema does not satisfy all the conditions of the 3NF because the functional dependency represented by Transport\_company and Mode\_of\_Transportation is a functional dependency where neither Transport\_company is a super key nor Mode\_of\_Transportation is a prime attribute. Thus, further decomposition is required.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Departure\_Time | Departure\_Date | Departure\_City | Departure\_State | Departure\_Country | Departure\_Planet | Price | Ticket\_type | Is\_booked |

TICKET-A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket\_number | Baggage\_weight | Seat\_number | Destination\_Time | Destination\_Date |

**FD1**

|  |  |  |  |
| --- | --- | --- | --- |
| Destination\_City | Destination\_State | Destination\_Country | Destination\_Planet |

**FD2**

TICKET-B

|  |  |
| --- | --- |
| Mode\_of\_Transportation | Transport\_company |

1. The **TICKET** relation schema satisfies the BCNF because there exists no functional dependency X 🡪 A where A is not a prime attribute and X is not a super key.
2. **HOTEL**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Hotel\_id | Hotel\_name | Street\_Number | City | State | Country | Planet |

|  |  |  |  |
| --- | --- | --- | --- |
| Star\_rating | Number\_of\_Meals | Price | is\_Partner |

**FD1**

1. The **HOTEL** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **HOTEL** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key Hotel\_id.
3. The **HOTEL** relation schema satisfies the 3NF because Hotel\_id is the superkey of that table (at least one of the conditions is satisfied).
4. The **HOTEL** relation schema satisfies the BCNF because Hotel\_id is the superkey of that table.
5. **ROOM**

**FD1**

|  |  |  |  |
| --- | --- | --- | --- |
| Room\_Number | H\_id | Capacity | Room\_Type |

1. The **ROOM** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **ROOM** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key consisting of Room\_Number and H\_id.
3. The **ROOM** relation schema satisfies the 3NF because Room\_Number and H\_id are the superkeys of that table (at least one of the conditions is satisfied).
4. The **ROOM** relation schema satisfies the BCNF because Room\_Number and H\_id are the superkeys of that table.
5. **CONTRACT**

**FD1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Contract\_id | is\_Refundable | is\_Signed | Expiry\_date | Travel\_reason | Employee\_ssn | Psprt\_id |

1. The **CONTRACT** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **CONTRACT** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key Contract\_id.
3. The **CONTRACT** relation schema satisfies the 3NF because Contract \_id is the superkey of that table (at least one of the conditions is satisfied).
4. The **CONTRACT** relation schema satisfies the BCNF because Contract \_id is the superkey of that table.
5. **COMPANION**

**FD1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | P\_id | Chat\_id | Date\_of\_Birth | Gender | Relationship |

1. The **COMPANION** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **COMPANION** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key consisting of First\_Name, Middle\_Name, Last\_Name, and P\_id.
3. The **COMPANION** relation schema satisfies the 3NF because First\_Name, Middle\_Name, Last\_Name, and P\_id are the superkeys of that table (at least one of the conditions is satisfied).
4. The **COMPANION** relation schema satisfies the BCNF because because First\_Name, Middle\_Name, Last\_Name, and P\_id are the superkeys of that table.
5. **BILL**

**FD1**

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction\_id | Payment\_date | Method\_of\_Payment | Psprt\_id |

1. The **BILL** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **BILL** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key Transaction\_id.
3. The **BILL** relation schema satisfies the 3NF because Transaction \_id is the superkey of that table (at least one of the conditions is satisfied).
4. The **BILL** relation schema satisfies the BCNF because Transaction \_id is the superkey of that table.

**FD1**

1. **DEPARTMENT**

|  |  |  |
| --- | --- | --- |
| Department\_Name | Branch\_Nb | Description |

1. The **DEPARTMENT** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **DEPARTMENT** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key consisting of Department\_Name and Branch\_Nb.
3. The **DEPARTMENT** relation schema satisfies the 3NF because Department\_Name and Branch\_Nb are the superkeys of that table (at least one of the conditions is satisfied).
4. The **DEPARTMENT** relation schema satisfies the BCNF because Department\_Name and Branch\_Nb are the superkeys of that table.

**FD1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | P\_id | Chat\_id | Relationship |

1. **EMERGENCY\_CONTACT1**
2. The **EMERGENCY\_CONTACT1** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
3. The **EMERGENCY\_CONTACT1** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key consisting of First\_Name, Middle\_Name, Last\_Name, and P\_id.
4. The **EMERGENCY\_CONTACT1** relation schema satisfies the 3NF because First\_Name, Middle\_Name, Last\_Name, and P\_id are the superkeys of that table (at least one of the conditions is satisfied).
5. The **EMERGENCY\_CONTACT1** relation schema satisfies the BCNF because First\_Name, Middle\_Name, Last\_Name, and P\_id are the superkeys of that table of that table.
6. **EMERGENCY\_CONTACT2**

**FD1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| First\_Name | Middle\_Name | Last\_Name | essn | Chat\_id | Relationship |

1. The **EMERGENCY\_CONTACT2** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **EMERGENCY\_CONTACT2** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key consisting of First\_Name, Middle\_Name, Last\_Name, and essn.
3. The **EMERGENCY\_CONTACT2** relation schema satisfies the 3NF because First\_Name, Middle\_Name, Last\_Name, and essn are the superkeys of that table (at least one of the conditions is satisfied).
4. The **EMERGENCY\_CONTACT2** relation schema satisfies the BCNF because First\_Name, Middle\_Name, Last\_Name, essn are the superkeys of that table of that table
5. **BRANCH**

**FD1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Branch\_Number | Street\_Number | City | State | Country | Planet |

1. The **BRANCH** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **BRANCH** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key Branch\_Number.
3. The **BRANCH** relation schema satisfies the 3NF because Branch\_Number is the superkey of that table (at least one of the conditions is satisfied).
4. The **BRANCH** relation schema satisfies the BCNF because Branch\_Number is the superkey of that table.
5. **PARTNER**

**FD1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Partner\_id | Name | City | State | Country | Street\_Number | Service | Planet |

1. The **PARTNER** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
2. The **PARTNER** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key Partner \_id.
3. The **PARTNER** relation schema satisfies the 3NF because Partner\_id is the superkey of that table (at least one of the conditions is satisfied).
4. The **PARTNER** relation schema satisfies the BCNF because Partner\_id is the superkey of that table.
5. **CUSTOMER\_PURCHASE\_HISTORY**

|  |  |
| --- | --- |
| Purchased\_Tickets\_id | Psprt\_id |

* + The **CUSTOMER\_PURCHASE\_HISTORY** relation schema only includes its primary key, which consists of Purchase\_Ticket\_id and Psprt\_id. Therefore, this table naturally satisfies the 1NF, 2NF, 3NF, and BCNF.

1. **PARTNER\_OPERATING\_COUNTRIES**

|  |  |
| --- | --- |
| Country | Prtnr\_id |

* + The **PARTNER\_OPERATING\_COUNTRIES** relation schema only includes its primary key, which consists of Purchase\_Ticket\_id and Psprt\_id. Therefore, this table naturally satisfies the 1NF, 2NF, 3NF, and BCNF.

1. **RECOMMENDS**

|  |  |  |
| --- | --- | --- |
| Employee\_ssn | Psprt\_id | Htl\_id |

* + The **RECOMMENDS** relation schema only includes its primary key, which consists of Employee\_ssn, Psprt\_id, and Htl\_id. Therefore, this table naturally satisfies the 1NF, 2NF, 3NF, and BCNF.

1. **MANAGES**

|  |  |
| --- | --- |
| Employee\_ssn | Psprt\_id |

* + The **MANAGES** relation schema only includes its primary key, which consists of Employee\_ssn and Psprt\_id. Therefore, this table naturally satisfies the 1NF, 2NF, 3NF, and BCNF.

1. **References[[4]](#footnote-4)**

El Masri, R., & Navanthe, S. B. (n.d.). *Database Systems: Models, Languages, Design, and Application Programming.* Pearson.

Kelly, S. M. (2021, October 29). Facebook changes its company name to Meta. *CNN*.

Microsoft Corporation. (1992). *Microsoft Visio*. Retrieved from https://www.microsoft.com/en-us/store/collections/visio

Zenou, T. (2022, June 30). A novel predicted the metaverse (and hyperinflation) 30 years ago. *The Washington Post*.

1. Visio was used to create the ER Diagram. [↑](#footnote-ref-1)
2. Latin phrase meaning “to the stars” [↑](#footnote-ref-2)
3. There has been an American ban on Middle East Airlines (MEA) since the hijacking of a Trans World Airlines (TWA) American Passenger plane at Beirut Airport in 1985. Consequently, the airport was placed on the US terror list. [↑](#footnote-ref-3)
4. References were cited using APA style. [↑](#footnote-ref-4)