Group 1

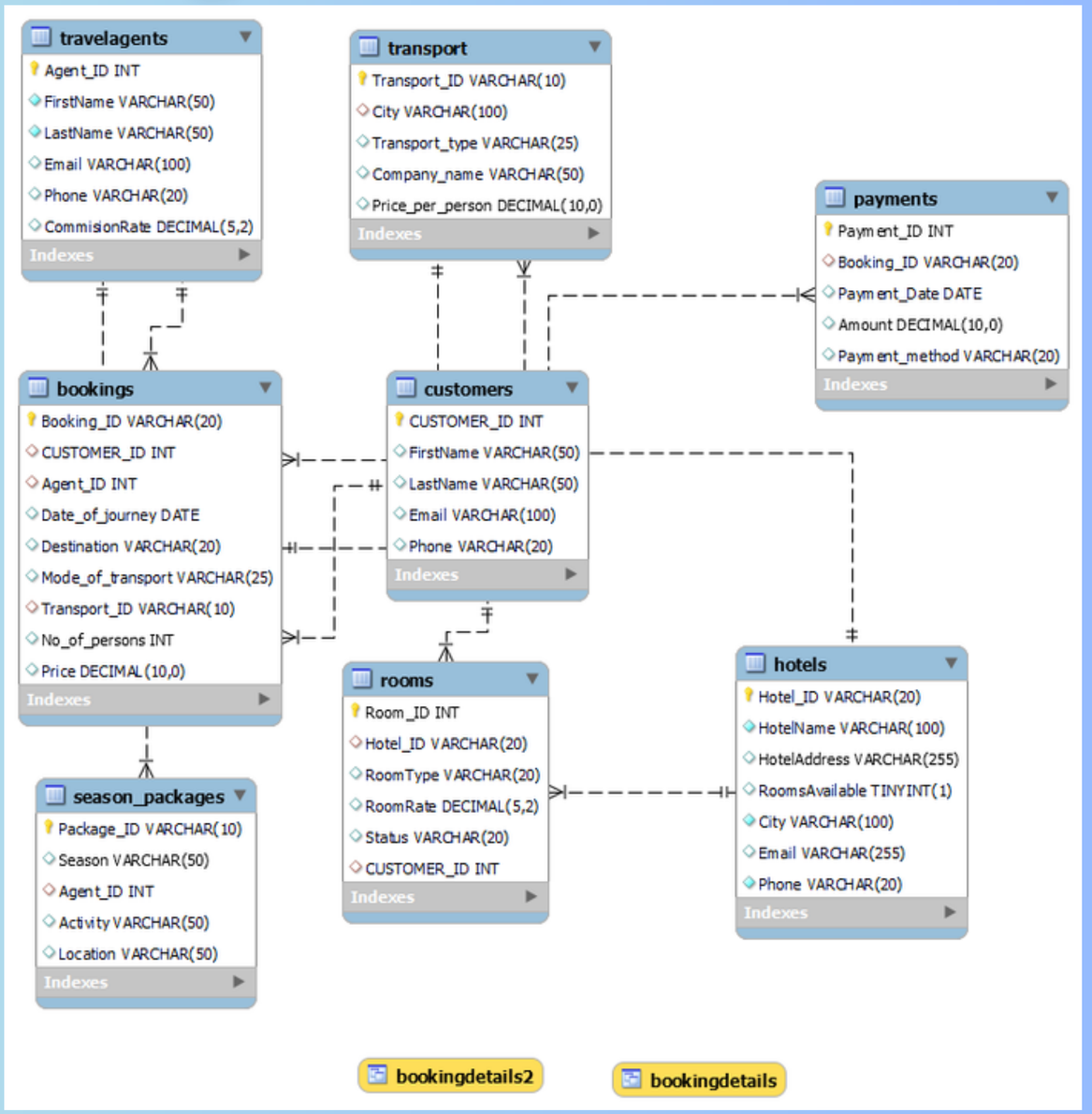
Aasifa Noor - Abinayaa B - Emilia Morales Vega - Ketana K - Rhona B

Summary

Our database is designed to be a centralized repository of information for Getaways.com the travel agency.

It aims to provide a reliable and efficient system for managing and retrieving travel-related data, which can be used by travel agents for planning custom itineraries.

The user-friendly database allows agents to easily search, filter, and sort through travel data, It will also provide real-time updates to ensure access to the most up-to-date information.



Core & Advanced Requirements

CORE Requirements

- **Create a view using JOINS** - *Abinayaa B*
- **Create a stored function** - *Emilia MV*
- **Provide example query with a subquery to demonstrate how to extract data from your DB for analysis** - *Aasifa Noor*
- **Create DB Diagram** - *Abinayaa B*
- **Create presentation** - *Emilia MV*

Advanced Requirements

- **Create a view that uses at least 3-4 base tables** - *Abinayaa B*
- **Prepare and demonstrate a query that uses the view to produce a logically arranged result set for analysis** - *Emilia MV*
- **Create a trigger** - *Ketana*
- **Create an event** - *Ketana*
- **Create stored procedure** - *Rhona B*
- **Prepare an example query with GROUP BY and HAVING** - *Rhona B*

Using Joins to Create a View

```
CREATE VIEW BookingDetails AS
SELECT
    B.Booking_ID,
    B.Date_of_journey,
    B.Destination,
    B.Mode_of_transport,
    B.No_of_persons,
    B.Price,
    C.Customer_ID,
    C.FirstName AS Customer_FirstName,
    C.LastName AS Customer_LastName,
    C.Email AS Customer_Email,
    C.Phone AS Customer_Phone,
    T.Agent_ID,
    T.FirstName AS Agent_FirstName,
    T.LastName AS Agent_LastName,
    T.Email AS Agent_Email,
    T.Phone AS Agent_Phone
FROM
    Bookings B
    JOIN Customers C ON B.CUSTOMER_ID = C.Customer_ID
    JOIN TravelAgents T ON B.Agent_ID = T.Agent_ID;
```

Table 1	Table 2	Join Type	Join Condition
Bookings	Customers	JOIN	B.CUSTOMER_ID = C.CUSTOMER_ID
Bookings	TravelAgents	JOIN	B.Agent_ID = T.Agent_ID

Date_of_journey	Destination	Mode_of_transpor	No_of_persons	Price	Customer_ID	Customer_FirstName	Customer_LastName	Customer_Email	Customer_Phone	Agent_ID	Agent_FirstName	Agent_LastName	Agent_Email	Agent_Phone
15/01/2024	New York	Air	3	1250	8	Sophia	Lee	sophia.lee@example.com	999-888-7777	101	Adam	Smith	smith.adam@getaways.com	07485266890
12/02/2024	Cairo	Air	2	850	6	Eva	Miller	eva.miller@example.com	777-888-9999	102	Naushad	Khan	khan.naushad@getaways.com	07265893251
01/07/2024	London	Eurorail	1	300	2	Jane	Smith	jane.smith@example.com	987-654-3210	103	Daniel	Hood	hood.daniel@getaways.com	07389456219
25/06/2024	New York	Air	2	650	3	Bob	Johnson	bob.johnson@example.com	555-123-4567	103	Daniel	Hood	hood.daniel@getaways.com	07389456219
28/10/2023	Mexico City	Air	5	1000	5	Charlie	Brown	charlie.brown@example.cor	111-222-3333	104	Abigail	Lawrence	lawrence.abigail@getaways.co	07534210890
09/06/2024	Amsterdam	Eurorail	2	900	1	John	Doe	john.doe@example.com	123-456-7890	105	Roma	Shood	shood.roma@getaways.com	07563210089
30/08/2024	Quito	Air	1	400	9	Daniel	Wilson	daniel.wilson@example.cor	222-333-4444	105	Roma	Shood	shood.roma@getaways.com	07563210089

Stored Function

A stored function is a set of SQL statements that perform a specific task and are stored in the database server. The primary purpose of a stored function is to encapsulate a specific operation or computation that can be reused within SQL queries, procedures, or triggers.

```
DELIMITER $$


CREATE FUNCTION GetTotalBookingsPerAgent(agentID INT) RETURNS INT DETERMINISTIC
BEGIN
    DECLARE totalBookings INT;

    SELECT COUNT(*) INTO totalBookings
    FROM Bookings
    WHERE Agent_ID = agentID;

    RETURN totalBookings;
END $$

DELIMITER ;

SELECT GetTotalBookingsPerAgent(103); -- Change Agent_ID
```

	GetTotalBookingsPerAgent(103)
▶	2

Example query with a subquery to demonstrate how to extract data for analysis

Query to fetch Agent First name & surname, with commission against bookings in Descending order.

```
SELECT f.FirstName, f.LastName, Commission
from (SELECT c.Agent_ID, sum(commission) as commission from
(SELECT a.agent_id, (CommisionRate * price) AS commission
  FROM travelagents as a,
  bookings as b
  where a.agent_id = b.agent_id) as c
  group by c.agent_id) as d
  , travelagents f
where d.agent_id = f.agent_id
order by commission desc;
```

	FirstName	LastName	Commission
▶	Adam	Smith	125.00
	Roma	Shood	104.00
	Daniel	Hood	95.00
	Naushad	Khan	85.00
	Abigail	Lawrence	80.00

Example query with a subquery to demonstrate how to extract data for analysis

Query to fetch Transport and Non-Transport cost to analyse Customer preference on Transport and Non-transport expenses.

```
SELECT C.TRANSPORT_ID,
            C.Booking_ID,
            (c.PRICE - TC) as Non_Transport_Cost,
            TC as Transport_cost
from (SELECT a.transport_id, (No_of_persons * Price_per_person) as tc
from TRANSPORT A,
    BOOKINGS b
WHERE A.TRANSPORT_ID = B.TRANSPORT_ID) as d
    , bookings as c
where d.transport_id = c.transport_id
order by TRANSPORT_COST DESC;
```

TRANSPORT_ID	BOOKING_ID	Non_Transport_Cost	Transport_cost
MX17	MX010	150	850
NY11	NY005	-100	750
NY11	NY015	500	750
NY11	NY005	150	500
NY11	NY015	750	500
CI8	CAI024	650	200
QU14	QUI011	250	150
AM3	ADM021	850	50
LD21	LON008	280	20

Using Joins to Create a 3-4 table View

```
CREATE VIEW BookingDetails AS
SELECT
    B.Booking_ID,
    B.Date_of_journey,
    B.Destination,
    B.Mode_of_transport,
    B.No_of_persons,
    B.Price,
    C.Customer_ID,
    C.FirstName AS Customer_FirstName,
    C.LastName AS Customer_LastName,
    C.Email AS Customer_Email,
    C.Phone AS Customer_Phone,
    T.Agent_ID,
    T.FirstName AS Agent_FirstName,
    T.LastName AS Agent_LastName,
    T.Email AS Agent_Email,
    T.Phone AS Agent_Phone
FROM
    Bookings B
    JOIN Customers C ON B.CUSTOMER_ID = C.Customer_ID
    JOIN TravelAgents T ON B.Agent_ID = T.Agent_ID;
```

Table 1	Table 2	Join Type	Join Condition
Bookings	Customers	JOIN	B.CUSTOMER_ID = C.CUSTOMER_ID
Bookings	TravelAgents	JOIN	B.Agent_ID = Ta.Agent_ID
Bookings	Season Packages	LEFT JOIN	b.Agent_ID = sp.Agent_ID

Booking_ID	Customer_ID	CustomerName	Agent_ID	AgentName	Date_of_journey	Destination	Mode_of_transport	No_of_persons	Price	Package_ID	Season	Activity	Location
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG1	WINTER	Adventure	Amsterdam
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG5	SUMMER	Safari	Quito
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG7	SPRING	Adventure	Cairo
CAI024	6	Eva Miller	102	Naushad Khan	12/02/2024	Cairo	Air	2	850	PKG2	WINTER	Luxury	New York
CAI024	6	Eva Miller	102	Naushad Khan	12/02/2024	Cairo	Air	2	850	PKG8	SPRING	Luxury	London
LON008	2	Jane Smith	103	Daniel Hood	01/07/2024	London	Eurorail	1	300	PKG6	SUMMER	Spa	Denpasar
NY005	3	Bob Johnson	103	Daniel Hood	25/06/2024	New York	Air	2	650	PKG6	SUMMER	Spa	Denpasar
MX010	5	Charlie Brown	104	Abigail Lawrence	28/10/2023	Mexico City	Air	5	1000	PKG3	AUTUMN	Spa	Mexico City
ADM021	1	John Doe	105	Roma Shood	09/06/2024	Amsterdam	Eurorail	2	900	PKG4	AUTUMN	Luxury	Denpasar

Analysis using view - Query

```
SELECT
    Booking_ID,
    Customer_ID,
    CustomerName,
    Agent_ID,
    AgentName,
    Date_of_journey,
    Destination,
    Mode_of_transport,
    No_of_persons,
    Price,
    Package_ID,
    Season,
    Activity,
    Location
FROM
    BookingDetails2
ORDER BY
    Date_of_journey;
```

Booking_ID	Customer_ID	CustomerName	Agent_ID	AgentName	Date_of_journey	Destination	Mode_of_transport	No_of_persons	Price	Package_ID	Season	Activity	Location
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG1	WINTER	Adventure	Amsterdam
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG5	SUMMER	Safari	Quito
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG7	SPRING	Adventure	Cairo
CAI024	6	Eva Miller	102	Naushad Khan	12/02/2024	Cairo	Air	2	850	PKG2	WINTER	Luxury	New York
CAI024	6	Eva Miller	102	Naushad Khan	12/02/2024	Cairo	Air	2	850	PKG8	SPRING	Luxury	London
LON008	2	Jane Smith	103	Daniel Hood	01/07/2024	London	Eurorail	1	300	PKG6	SUMMER	Spa	Denpasar
NY005	3	Bob Johnson	103	Daniel Hood	25/06/2024	New York	Air	2	650	PKG6	SUMMER	Spa	Denpasar
MX010	5	Charlie Brown	104	Abigail Lawrence	28/10/2023	Mexico City	Air	5	1000	PKG3	AUTUMN	Spa	Mexico City
ADM021	1	John Doe	105	Roma Shood	09/06/2024	Amsterdam	Eurorail	2	900	PKG4	AUTUMN	Luxury	Denpasar

	Booking_ID	Customer_ID	CustomerName	Agent_ID	AgentName	Date_of_journey	Destination	Mode_of_transport	No_of_persons	Price	Package_ID	Season	Activity	Location
▶	MX010	5	Charlie Brown	104	Abigail Lawrence	2023-10-28	Mexico City	Air	5	1000	PKG3	AUTUMN	Spa	Mexico City
	NY015	8	Sophia Lee	101	Adam Smith	2024-01-15	New York	Air	3	1250	PKG1	WINTER	Adventure	Amsterdam
	NY015	8	Sophia Lee	101	Adam Smith	2024-01-15	New York	Air	3	1250	PKG5	SUMMER	Safari	Quito
	NY015	8	Sophia Lee	101	Adam Smith	2024-01-15	New York	Air	3	1250	PKG7	SPRING	Adventure	Cairo
	CAI024	6	Eva Miller	102	Naushad Khan	2024-02-12	Cairo	Air	2	850	PKG2	WINTER	Luxury	New York
	CAI024	6	Eva Miller	102	Naushad Khan	2024-02-12	Cairo	Air	2	850	PKG8	SPRING	Luxury	London
	ADM021	1	John Doe	105	Roma Shood	2024-06-09	Amsterdam	Eurorail	2	900	PKG4	AUTUMN	Luxury	Denpasar
	NY005	3	Bob Johnson	103	Daniel Hood	2024-06-25	New York	Air	2	650	PKG6	SUMMER	Spa	Denpasar
	LON008	2	Jane Smith	103	Daniel Hood	2024-07-01	London	Eurorail	1	300	PKG6	SUMMER	Spa	Denpasar
	QUI011	9	Daniel Wilson	105	Roma Shood	2024-08-30	Quito	Air	1	400	PKG4	AUTUMN	Luxury	Denpasar

Analysis using view - Tableu

Booking_ID	Customer_ID	CustomerName	Agent_ID	AgentName	Date_of_journey	Destination	Mode_of_transport	No_of_persons	Price	Package_ID	Season	Activity	Location
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG1	WINTER	Adventure	Amsterdam
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG5	SUMMER	Safari	Quito
NY015	8	Sophia Lee	101	Adam Smith	15/01/2024	New York	Air	3	1250	PKG7	SPRING	Adventure	Cairo
CAI024	6	Eva Miller	102	Naushad Khan	12/02/2024	Cairo	Air	2	850	PKG2	WINTER	Luxury	New York
CAI024	6	Eva Miller	102	Naushad Khan	12/02/2024	Cairo	Air	2	850	PKG8	SPRING	Luxury	London
LON008	2	Jane Smith	103	Daniel Hood	01/07/2024	London	Eurorail	1	300	PKG6	SUMMER	Spa	Denpasar
NY005	3	Bob Johnson	103	Daniel Hood	25/06/2024	New York	Air	2	650	PKG6	SUMMER	Spa	Denpasar
MX010	5	Charlie Brown	104	Abigail Lawrence	28/10/2023	Mexico City	Air	5	1000	PKG3	AUTUMN	Spa	Mexico City
ADM021	1	John Doe	105	Roma Shood	09/06/2024	Amsterdam	Eurorail	2	900	PKG4	AUTUMN	Luxury	Denpasar

	Season / Package ID							
	AUTUMN		SPRING		SUMMER		WINTER	
Agent ID	PKG3	PKG4	PKG7	PKG8	PKG5	PKG6	PKG1	PKG2
101			1,250		1,250		1,250	
102				850				850
103						950		
104	1,000							
105		1,300						

	Package ID / Destination							
	PKG1	PKG2	PKG3	PKG4		PKG5	PKG6		PKG7	PKG8
Customer ID	New York	Cairo	Mexico City	Amster..	Quito	New York	London	New York	New York	Cairo
1				2						
2							1			
3								2		
5			5							
6		2								2
8	3					3			3	
9					1					

Trigger & Event

Trigger

A trigger is the occurrence of a specific event or action that prompts a response or action in the database.

```
DELIMITER //
CREATE TRIGGER AfterBookingInsert
AFTER INSERT ON Bookings
FOR EACH ROW
BEGIN
   UPDATE Rooms
   SET Status = 'Booked'
   WHERE CUSTOMER_ID = NEW.CUSTOMER_ID;
 END;
//
DELIMITER ;
```

Event

An event refers to a specific action or occurrence that triggers a response or action in the database, such as a user making a booking or updating their travel preferences.

```
CREATE EVENT RoomUpdate
ON COMPLETION PRESERVE
DO BEGIN UPDATE Rooms
SET Status = 'Available';
END;
```

Stored Procedure

```
DELIMITER //
CREATE PROCEDURE GetHotelInfo(IN hotelID
VARCHAR(20))
BEGIN
    SELECT *
  FROM Hotels
  WHERE Hotel_ID = hotelID;
  END//
DELIMITER ;


CALL GetHotelInfo ('SRE');
```

A stored procedure is a set of SQL statements that perform a specific task or a series of tasks and are stored in a database. Stored procedures are named and stored in the database, and they can be invoked or called by other programs or scripts.

Hotel_ID	HotelName	HotelAddress	RoomsAvailable	City	Email	Phone
SRE	Sunset Resort	Orizaba 95, Roma Norte, CDMX 6700,	1	Mexico City	reservaciones@sunsetresort.com	+552345678

Prepare an Example Query with GROUP BY and HAVING

Query to find hotels with an average room rate greater than 60

```
Select
H. Hotel_ID,
H. HotelName,
AVG (R.RoomRate) AS AvgRoomRate
From
Hotels H
JOIN Rooms R ON H. Hotel_ID = R.Hotel_ID
GROUP BY
H. Hotel_ID, H.HotelName
HAVING
AVG(R.RoomRate) > 60.00;
```

Hotel_ID	HotelName	AvgRoomRate
CLI	City Lights Inn	85.000000
ESR	Eternal Sunshine Resort	62.500000
GPH	Grand Plaza Hotel	250.000000
URH	Urban Retreat Hotel	375.000000

Q&A TIME

We hope you enjoyed our presentation!