

# SINDH MADRESSATUL ISLAM UNIVERSITY

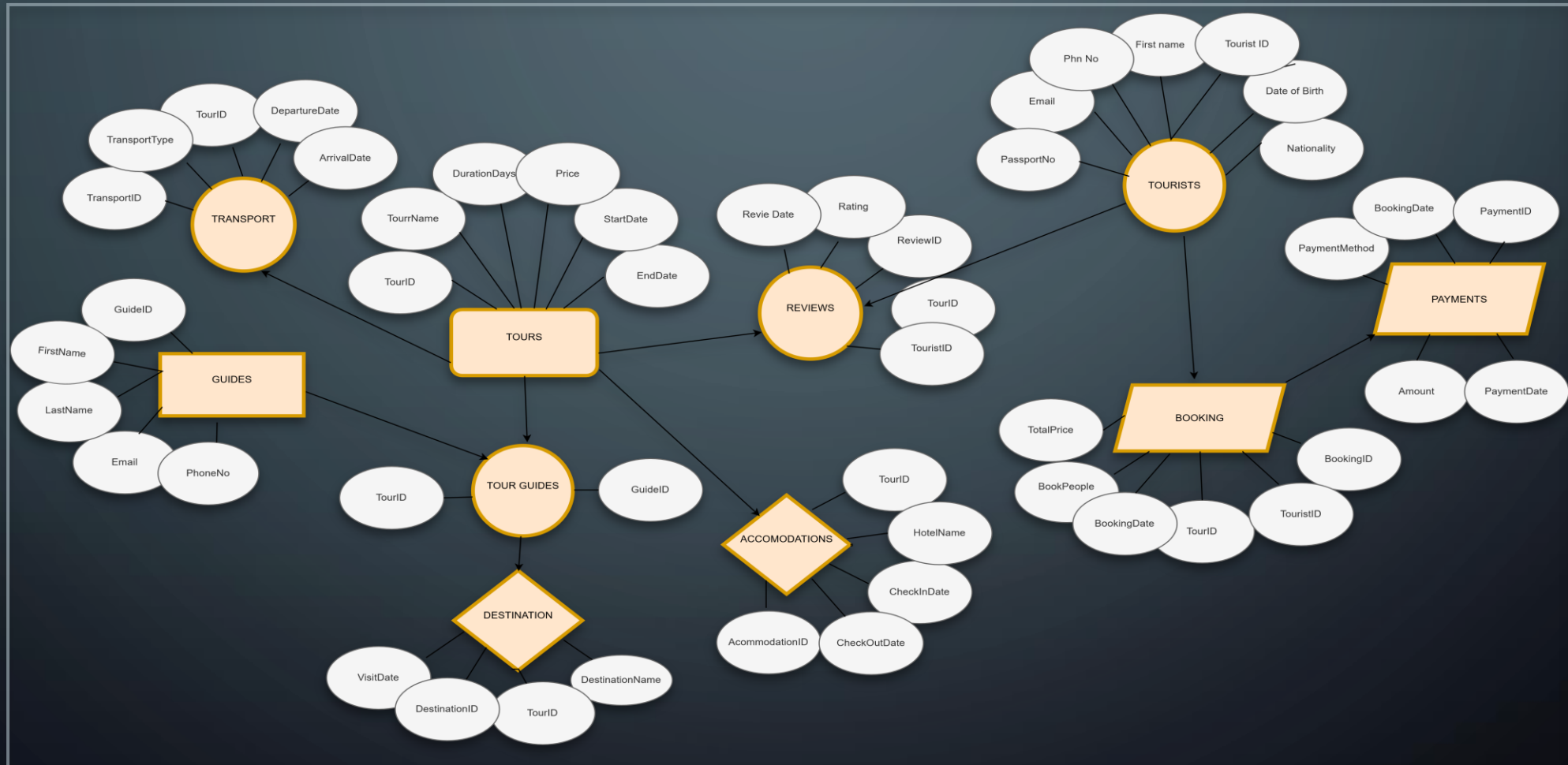


TOPIC NAME: TOURISM MANAGEMENT  
DEPARTMENT OF SOFTWARE ENGINEERING

## GROUP MEMBERS:

AQSA HANIF (BSE-23F-138)  
HADIYA MUNIR (BSE-23F-147)  
MUTAIBA ALI (BSE-23F-158)

# ER MODEL OF TOURISM MANAGEMENT:



# WHAT IS TOURISM MANAGEMENT AND HOW IT WORKS:

## TOURISM MANAGEMENT:

Tourism management involves overseeing and organizing various aspects of the tourism industry, such as planning, marketing, operations, and customer service. It focuses on managing destinations, attractions, accommodations, transportation, and activities to ensure a positive experience for travelers.

## HOW IT WORKS:

In tourism management, businesses use databases to keep track of all the important information like customer details, bookings, accommodations, and attractions. These databases help in organizing data efficiently and making it easier to access when needed. Additionally, tourism management databases allow businesses to run queries to extract specific data such as customer preferences, popular destinations, revenue reports, and booking patterns.

# CONTRIBUTION OF EACH MEMBER:

## MUTAIBA ALI:

```
Schema - Data Insertion & Query
Limit to 1000 rows
159 -- Query 1: Selects the first name, last name, and email of tourists who have booked TourID 1.
160 • SELECT T.FirstName, T.LastName, T.Email
161 FROM Tourists T
162 JOIN Bookings B ON T.TouristID = B.TouristID
163 WHERE B.TourID = 1;
164
165 -- Query 2: Calculates the total revenue generated from TourID 1.
166 • SELECT SUM(B.TotalPrice) AS TotalRevenue
167 FROM Bookings B
168 WHERE B.TourID = 1;
169
170 -- Query 3: Retrieves the name and average rating of each tour.
171 • SELECT TourName, AVG(Rating) AS AverageRating
172 FROM Tours T
173 JOIN Reviews R ON T.TourID = R.TourID
174 GROUP BY T.TourID, T.TourName;
175
176 -- Query 4: Selects the first name, last name, and email of guides assigned to TourID 1.
177 • SELECT G.FirstName, G.LastName, G.Email
178 FROM Guides G
179 JOIN TourGuides TG ON G.GuideID = TG.GuideID
180 WHERE TG.TourID = 1;
181
182 -- Query 5: Retrieves details of the booking with BookingID 1, including tourist information.
183 • SELECT B.BookingID, B.BookingDate, B.NumberOfPeople, B.TotalPrice, T.FirstName, T.LastName
184 FROM Bookings B
185 JOIN Tourists T ON B.TouristID = T.TouristID
186 WHERE B.BookingID = 1;
187
```

```
Schema - Data Insertion & Query
Limit to 1000 rows
188 -- Query 6: Selects the distinct tour names that have a perfect rating of 5.
189 • SELECT DISTINCT T.TourName
190 FROM Tours T
191 JOIN Reviews R ON T.TourID = R.TourID
192 WHERE R.Rating = 5;
193
194 -- Query 7: Retrieves details of destinations associated with TourID 1.
195 • SELECT D.DestinationName, D.VisitDate, D.Description
196 FROM Destinations D
197 WHERE D.TourID = 1;
198
199 -- Query 8: Counts the number of bookings each tourist has made.
200 • SELECT T.TouristID, T.FirstName, T.LastName, COUNT(B.BookingID) AS NumberOfBookings
201 FROM Tourists T
202 JOIN Bookings B ON T.TouristID = B.TouristID
203 GROUP BY T.TouristID, T.FirstName, T.LastName;
204
205 -- Query 9: Sums the total amount paid by each tourist.
206 • SELECT T.TouristID, T.FirstName, T.LastName, SUM(P.Amount) AS TotalAmountPaid
207 FROM Tourists T
208 JOIN Bookings B ON T.TouristID = B.TouristID
209 JOIN Payments P ON B.BookingID = P.BookingID
210 GROUP BY T.TouristID, T.FirstName, T.LastName;
211
212 -- Query 10: Retrieves the names of tours and their assigned guides.
213 • SELECT T.TourName, G.FirstName, G.LastName
214 FROM Tours T
215 JOIN TourGuides TG ON T.TourID = TG.TourID
216 JOIN Guides G ON TG.GuideID = G.GuideID;
```

```

218 -- Query 11: Selects the tour with the highest average rating.
219 • SELECT T.TourName, AVG(R.Rating) AS AverageRating
220 FROM Tours T
221 JOIN Reviews R ON T.TourID = R.TourID
222 GROUP BY T.TourID, T.TourName
223 ORDER BY AverageRating DESC
224 LIMIT 1;
225
226 -- Query 12: Calculates the total revenue from all bookings.
227 • SELECT SUM(TotalPrice) AS TotalRevenue
228 FROM Bookings;
229
230 -- Query 13: Counts the number of tourists for each tour.
231 • SELECT T.TourName, COUNT(B.BookingID) AS NumberOfTourists
232 FROM Tours T
233 JOIN Bookings B ON T.TourID = B.TourID
234 GROUP BY T.TourID, T.TourName;
235
236 -- Query 14: Counts the number of bookings each tourist has made, filtering to those with more than one booking.
237 • SELECT T.FirstName, T.LastName, COUNT(B.BookingID) AS NumberOfBookings
238 FROM Tourists T
239 JOIN Bookings B ON T.TouristID = B.TouristID
240 GROUP BY T.TouristID, T.FirstName, T.LastName
241 HAVING COUNT(B.BookingID) > 1;
242
243 -- Query 15: Selects reviews for TourID 1 along with the first name and last name of the tourists who wrote them.
244 • SELECT T.FirstName, T.LastName, R.Rating, R.Comment
245 FROM Reviews R
246 JOIN Tourists T ON R.TouristID = T.TouristID
247 WHERE R.TourID = 1;

```

```

248
249 -- Query 16: Calculates the average amount paid in all payments.
250 • SELECT AVG(P.Amount) AS AverageAmountPaid
251 FROM Payments P;
252
253 -- Query 17: Retrieves the names and start dates of tours starting in June.
254 • SELECT TourName, StartDate
255 FROM Tours
256 WHERE MONTH(StartDate) = 6;
257
258 -- Query 18: Selects the first name, last name, and nationality of tourists who are from the USA.
259 • SELECT FirstName, LastName, Nationality
260 FROM Tourists
261 WHERE Nationality = 'USA';
262
263 -- Query 19: Updates the phone number of the tourist with TouristID 1.
264 • UPDATE Tourists
265 SET PhoneNumber = '1234567899'
266 WHERE TouristID = 1;
267
268 -- Query 20: Updates the price of the tour with TourID 1.
269 • UPDATE Tours
270 SET Price = 1800.00
271 WHERE TourID = 1;
272

```

# AQSA HANIF:

```
272
273 -- Query 21: Updates the description of the destination with DestinationID 1.
274 • UPDATE Destinations
275 SET Description = 'A breathtaking park with a wide variety of wildlife.'
276 WHERE DestinationID = 1;
277
278 -- Query 22: Updates the comment of the review with ReviewID 1.
279 • UPDATE Reviews
280 SET Comment = 'Amazing experience! Highly recommend.'
281 WHERE ReviewID = 1;
282
283 -- Query 23: Selects all tours that have a duration greater than 5 days and a price less than $2000.
284 • SELECT * FROM Tours
285 WHERE DurationDays > 5 AND Price < 2000.00;
286
287 -- Query 24: Selects all bookings where the number of people is greater than 2.
288 • SELECT * FROM Bookings
289 WHERE NumberOfPeople > 2;
290
291 -- Query 25: Selects all reviews with a rating of 4 or higher.
292 • SELECT * FROM Reviews
293 WHERE Rating >= 4;
294
295 -- Query 26: Selects all payments made by credit card or PayPal.
296 • SELECT * FROM Payments
297 WHERE PaymentMethod = 'Credit Card' OR PaymentMethod = 'PayPal';
298
```

```
Limit to 1000 rows
298
299 -- Query 27: Selects all accommodations with a checkout date after July 7, 2024.
300 • SELECT * FROM Accommodations
301 WHERE CheckOutDate > '2024-07-07';
302
303 -- Query 28: Selects all flights departing between July 1, 2024, and July 31, 2024.
304 • SELECT * FROM Transport
305 WHERE TransportType = 'Flight' AND DepartureDate BETWEEN '2024-07-01' AND '2024-07-31';
306
307 -- Query 29: Selects all tourists who are from the USA or the UK.
308 • SELECT * FROM Tourists
309 WHERE Nationality = 'USA' OR Nationality = 'British';
310
311 -- Query 30: Counts the number of reviews each tourist has written.
312 • SELECT T.TouristID, T.FirstName, T.LastName, COUNT(R.ReviewID) AS NumberOfReviews
313 FROM Tourists T
314 JOIN Reviews R ON T.TouristID = R.TouristID
315 GROUP BY T.TouristID, T.FirstName, T.LastName;
316
317 -- Query 31: Sums the total number of people for each tour and selects the tour with the most people.
318 • SELECT T.TourName, SUM(B.NumberOfPeople) AS TotalPeople
319 FROM Tours T
320 JOIN Bookings B ON T.TourID = B.TourID
321 GROUP BY T.TourID, T.TourName
322 ORDER BY TotalPeople DESC
323 LIMIT 1;
```



```

Schema - Data Insertion & Query...
Limit to 1000 rows

325 -- Query 32: Counts the number of destinations for each tour.
326 • SELECT T.TourName, COUNT(D.DestinationID) AS NumberOfDestinations
327 FROM Tours T
328 JOIN Destinations D ON T.TourID = D.TourID
329 GROUP BY T.TourID, T.TourName;
330
331 -- Query 33: Selects the most expensive tour.
332 • SELECT TourName, Price
333 FROM Tours
334 ORDER BY Price DESC
335 LIMIT 1;
336
337 -- Query 34: Selects the first name and last name of tourists who have not written any reviews.
338 • SELECT T.FirstName, T.LastName
339 FROM Tourists T
340 LEFT JOIN Reviews R ON T.TouristID = R.TouristID
341 WHERE R.ReviewID IS NULL;
342
343 -- Query 35: Calculates the average number of people per booking.
344 • SELECT AVG(NumberOfPeople) AS AverageNumberOfPeople
345 FROM Bookings;
346
347 -- Query 36: Selects the shortest tour by duration.
348 • SELECT TourName, DurationDays
349 FROM Tours
350 ORDER BY DurationDays ASC
351 LIMIT 1;

```

```

SQL File 3*
Limit to 1000 rows

353 -- Query 37: Selects tourists and their booking dates for bookings made between April 1, 2024, and May 1, 2024.
354 • SELECT T.FirstName, T.LastName, B.BookingDate
355 FROM Tourists T
356 JOIN Bookings B ON T.TouristID = B.TouristID
357 WHERE B.BookingDate BETWEEN '2024-04-01' AND '2024-05-01';
358
359 -- Query 38: Sums the total amount paid for each tour.
360 • SELECT T.TourName, SUM(P.Amount) AS TotalAmountPaid
361 FROM Tours T
362 JOIN Bookings B ON T.TourID = B.TourID
363 JOIN Payments P ON B.BookingID = P.BookingID
364 GROUP BY T.TourID, T.TourName;
365
366 -- Query 39: Selects the first name, last name, and the number of tours each guide is assigned to, filtering to those with more than one tour.
367 • SELECT G.FirstName, G.LastName, COUNT(TG.TourID) AS NumberOfTours
368 FROM Guides G
369 JOIN TourGuides TG ON G.GuideID = TG.GuideID
370 GROUP BY G.GuideID, G.FirstName, G.LastName
371 HAVING COUNT(TG.TourID) > 1;
372
373 -- Query 40: Retrieves booking and payment details, including the payment method.
374 • SELECT B.BookingID, B.BookingDate, P.PaymentDate, P.Amount, P.PaymentMethod
375 FROM Bookings B
376 JOIN Payments P ON B.BookingID = P.BookingID;
377

```

# HADIYA MUNIR:

```
377
378 -- Query 41: Calculates the average rating for each tour.
379 • SELECT T.TourName, AVG(R.Rating) AS AverageRating
380 FROM Tours T
381 JOIN Reviews R ON T.TourID = R.TourID
382 GROUP BY T.TourID, T.TourName;
383
384 -- Query 42: Selects the start and end dates of all tours.
385 • SELECT TourName, StartDate, EndDate
386 FROM Tours;
387
388 -- Query 43: Counts the number of bookings for each month.
389 • SELECT MONTH(BookingDate) AS BookingMonth, COUNT(BookingID) AS NumberOfBookings
390 FROM Bookings
391 GROUP BY BookingMonth;
392
393 -- Query 44: Selects guides who are not assigned to any tours.
394 • SELECT G.FirstName, G.LastName
395 FROM Guides G
396 LEFT JOIN TourGuides TG ON G.GuideID = TG.GuideID
397 WHERE TG.TourID IS NULL;
398
399 -- Query 45: Counts the number of reviews each tour has received.
400 • SELECT T.TourName, COUNT(R.ReviewID) AS NumberOfReviews
401 FROM Tours T
402 JOIN Reviews R ON T.TourID = R.TourID
403 GROUP BY T.TourID, T.TourName;
404
```

```
404
405 -- Query 46: Counts the number of distinct tours each tourist has booked, filtering to those with more than one tour.
406 • SELECT T.FirstName, T.LastName, COUNT(DISTINCT B.TourID) AS NumberOfTours
407 FROM Tourists T
408 JOIN Bookings B ON T.TouristID = B.TouristID
409 GROUP BY T.TouristID, T.FirstName, T.LastName
410 HAVING COUNT(DISTINCT B.TourID) > 1;
411
412 -- Query 47: Retrieves destination details along with the tour name they belong to.
413 • SELECT D.DestinationName, T.TourName, D.VisitDate
414 FROM Destinations D
415 JOIN Tours T ON D.TourID = T.TourID;
416
417 -- Query 48: Calculates the average duration of all tours.
418 • SELECT AVG(DurationDays) AS AverageDuration
419 FROM Tours;
420
421 -- Query 49: Counts the number of tourists who have given a rating of 5.
422 • SELECT COUNT(DISTINCT R.TouristID) AS NumberOfTourists
423 FROM Reviews R
424 WHERE R.Rating = 5;
425
426 -- Query 50: Retrieves recent bookings (within the last month) along with tourist details.
427 • SELECT B.BookingID, B.BookingDate, T.FirstName, T.LastName
428 FROM Bookings B
429 JOIN Tourists T ON B.TouristID = T.TouristID
430 WHERE B.BookingDate > DATE_SUB(CURDATE(), INTERVAL 1 MONTH);
```



```

431
432 -- Query 51: Sums the total amount paid for bookings made in the last year.
433 • SELECT SUM(P.Amount) AS TotalAmountPaid
434 FROM Payments P
435 JOIN Bookings B ON P.BookingID = B.BookingID
436 WHERE B.BookingDate > DATE_SUB(CURDATE(), INTERVAL 1 YEAR);
437
438 -- Query 52: Selects distinct hotel names from accommodations.
439 • SELECT DISTINCT HotelName
440 FROM Accommodations;
441
442 -- Query 53: Counts the number of tourists from each nationality.
443 • SELECT Nationality, COUNT(TouristID) AS NumberOfTourists
444 FROM Tourists
445 GROUP BY Nationality;
446
447 -- Query 54: Selects distinct transport types from the transport table.
448 • SELECT DISTINCT TransportType
449 FROM Transport;
450
451 -- Query 55: Selects the longest tour by duration.
452 • SELECT TourName, DurationDays, StartDate, EndDate
453 FROM Tours
454 ORDER BY DurationDays DESC
455 LIMIT 1;
456

```

```

457 -- Query 56: Selects tours ending in June.
458 • SELECT TourName, EndDate
459 FROM Tours
460 WHERE MONTH(EndDate) = 6;
461 -- Query 57: Selects the first name, last name, and email of tourists who share a last name with other tourists.
462 • SELECT FirstName, LastName, Email
463 FROM Tourists
464 WHERE LastName IN (
465     SELECT LastName
466     FROM Tourists
467     GROUP BY LastName
468     HAVING COUNT(*) = 1
469 );
470 -- Query 58: Counts the number of bookings made by tourists younger than 30.
471 • SELECT COUNT(B.BookingID) AS NumberOfBookings
472 FROM Bookings B
473 JOIN Tourists T ON B.TouristID = T.TouristID
474 WHERE YEAR(CURDATE()) - YEAR(T.DateOfBirth) < 30;
475
476 -- Query 59: Counts the number of guides assigned to each tour.
477 • SELECT T.TourName, COUNT(TG.GuideID) AS NumberOfGuides
478 FROM Tours T
479 JOIN TourGuides TG ON T.TourID = TG.TourID
480 GROUP BY T.TourID, T.TourName;
481
482 -- Query 60: Retrieves the most recent booking details along with tourist information.
483 • SELECT B.BookingID, B.BookingDate, T.FirstName, T.LastName, B.NumberOfPeople, B.TotalPrice
484 FROM Bookings B
485 JOIN Tourists T ON B.TouristID = T.TouristID
486 ORDER BY B.BookingDate DESC
487 LIMIT 1;

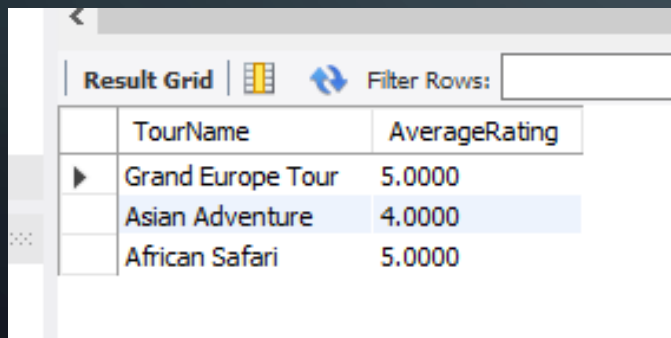
```

# SCHEMA, DATA INSERTION AND QUERIES:

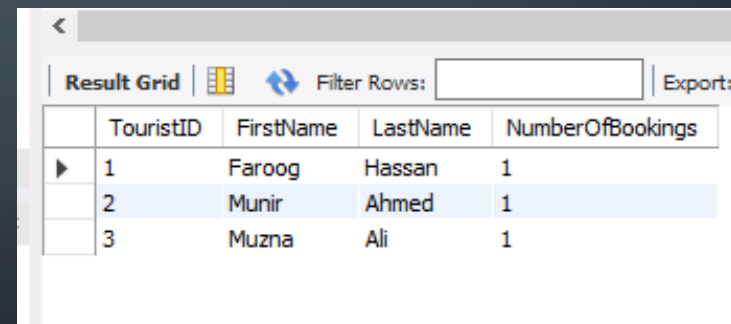
```
170 -- Query 3: Retrieves the name and average rating of each tour.
171 • SELECT TourName, AVG(Rating) AS AverageRating
172 FROM Tours T
173 JOIN Reviews R ON T.TourID = R.TourID
174 GROUP BY T.TourID, T.TourName;
175
```

```
176 -- Query 8: Counts the number of bookings each tourist has made.
177 • SELECT T.TouristID, T.FirstName, T.LastName, COUNT(B.BookingID) AS NumberOfBookings
178 FROM Tourists T
179 JOIN Bookings B ON T.TouristID = B.TouristID
180 GROUP BY T.TouristID, T.FirstName, T.LastName;
```

## OUTPUT:



	TourName	AverageRating
▶	Grand Europe Tour	5.0000
	Asian Adventure	4.0000
	African Safari	5.0000



	TouristID	FirstName	LastName	NumberOfBookings
▶	1	Faroog	Hassan	1
	2	Munir	Ahmed	1
	3	Muzna	Ali	1

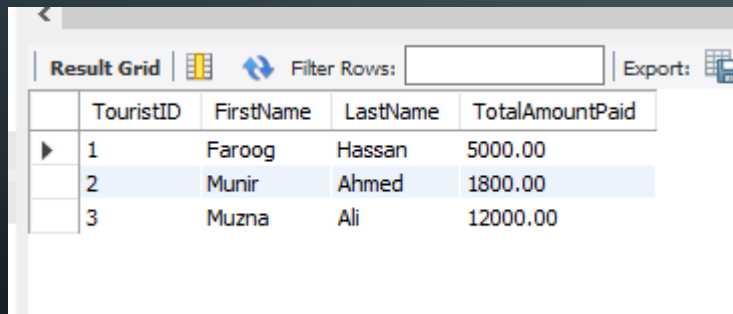
-- Query 9: Sums the total amount paid by each tourist.

- ```
SELECT T.TouristID, T.FirstName, T.LastName, SUM(P.Amount) AS TotalAmountPaid
FROM Tourists T
JOIN Bookings B ON T.TouristID = B.TouristID
JOIN Payments P ON B.BookingID = P.BookingID
GROUP BY T.TouristID, T.FirstName, T.LastName;
```

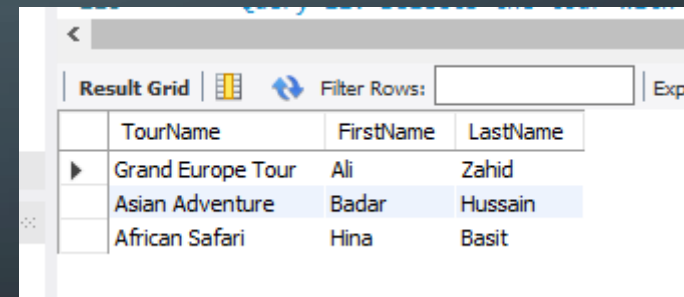
-- Query 10: Retrieves the names of tours and their assigned guides.

- ```
SELECT T.TourName, G.FirstName, G.LastName
FROM Tours T
JOIN TourGuides TG ON T.TourID = TG.TourID
JOIN Guides G ON TG.GuideID = G.GuideID;
```

## OUTPUT:



	TouristID	FirstName	LastName	TotalAmountPaid
▶	1	Faroog	Hassan	5000.00
	2	Munir	Ahmed	1800.00
	3	Muzna	Ali	12000.00



	TourName	FirstName	LastName
▶	Grand Europe Tour	Ali	Zahid
	Asian Adventure	Badar	Hussain
	African Safari	Hina	Basit

```
-- Query 30: Counts the number of reviews each tourist has written.
SELECT T.TouristID, T.FirstName, T.LastName, COUNT(R.ReviewID) AS NumberOfReviews
FROM Tourists T
JOIN Reviews R ON T.TouristID = R.TouristID
GROUP BY T.TouristID, T.FirstName, T.LastName;
```


## OUTPUT:


	TouristID	FirstName	LastName	NumberOfReviews
▶	1	Faroog	Hassan	1
	2	Munir	Ahmed	1
	3	Muzna	Ali	1

```
-- Query 33: Selects the most expensive tour.
SELECT TourName, Price
FROM Tours
ORDER BY Price DESC
LIMIT 1;
```

337 -- Query 34: Selects

Result Grid



 Filter Rows:

	TourName	Price
▶	African Safari	3000.00

```
-- Query 37: Selects tourists and their booking dates for bookings made between April 1, 2024, and May 1, 2024.
SELECT T.FirstName, T.LastName, B.BookingDate
FROM Tourists T
JOIN Bookings B ON T.TouristID = B.TouristID
WHERE B.BookingDate BETWEEN '2024-04-01' AND '2024-05-01';
```

```
-- Query 40: Retrieves booking and payment details, including the payment method.
SELECT B.BookingID, B.BookingDate, P.PaymentDate, P.Amount, P.PaymentMethod
FROM Bookings B
JOIN Payments P ON B.BookingID = P.BookingID;
```

## OUTPUT:

358

359 -- Query 38: Sums the total

Result Grid

	FirstName	LastName	BookingDate
	Faroog	Hassan	2024-04-01
	Munir	Ahmed	2024-05-01

Result Grid

Filter Rows:

Export: Wrap Cell Content:

	BookingID	BookingDate	PaymentDate	Amount	PaymentMethod
	1	2024-04-01	2024-04-02	5000.00	Credit Card
	2	2024-05-01	2024-05-02	1800.00	PayPal
	3	2024-06-01	2024-06-02	12000.00	Bank Transfer

```
-- Query 53: Counts the number of tourists from each nationality.  
SELECT Nationality, COUNT(TouristID) AS NumberOfTourists  
FROM Tourists  
GROUP BY Nationality;
```

```
-- Query 60: Retrieves the most recent booking details along with tourist information.  
SELECT B.BookingID, B.BookingDate, T.FirstName, T.LastName, B.NumberOfPeople, B.TotalPrice  
FROM Bookings B  
JOIN Tourists T ON B.TouristID = T.TouristID  
ORDER BY B.BookingDate DESC  
LIMIT 1;
```

## OUTPUT:

	Nationality	NumberOfTourists
▶	USA	1
	Canada	1
	UK	1

	BookingID	BookingDate	FirstName	LastName	NumberOfPeople	TotalPrice
▶	3	2024-06-01	Muzna	Ali	4	12000.00



The background is a dark blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing components.

# THANK YOU!