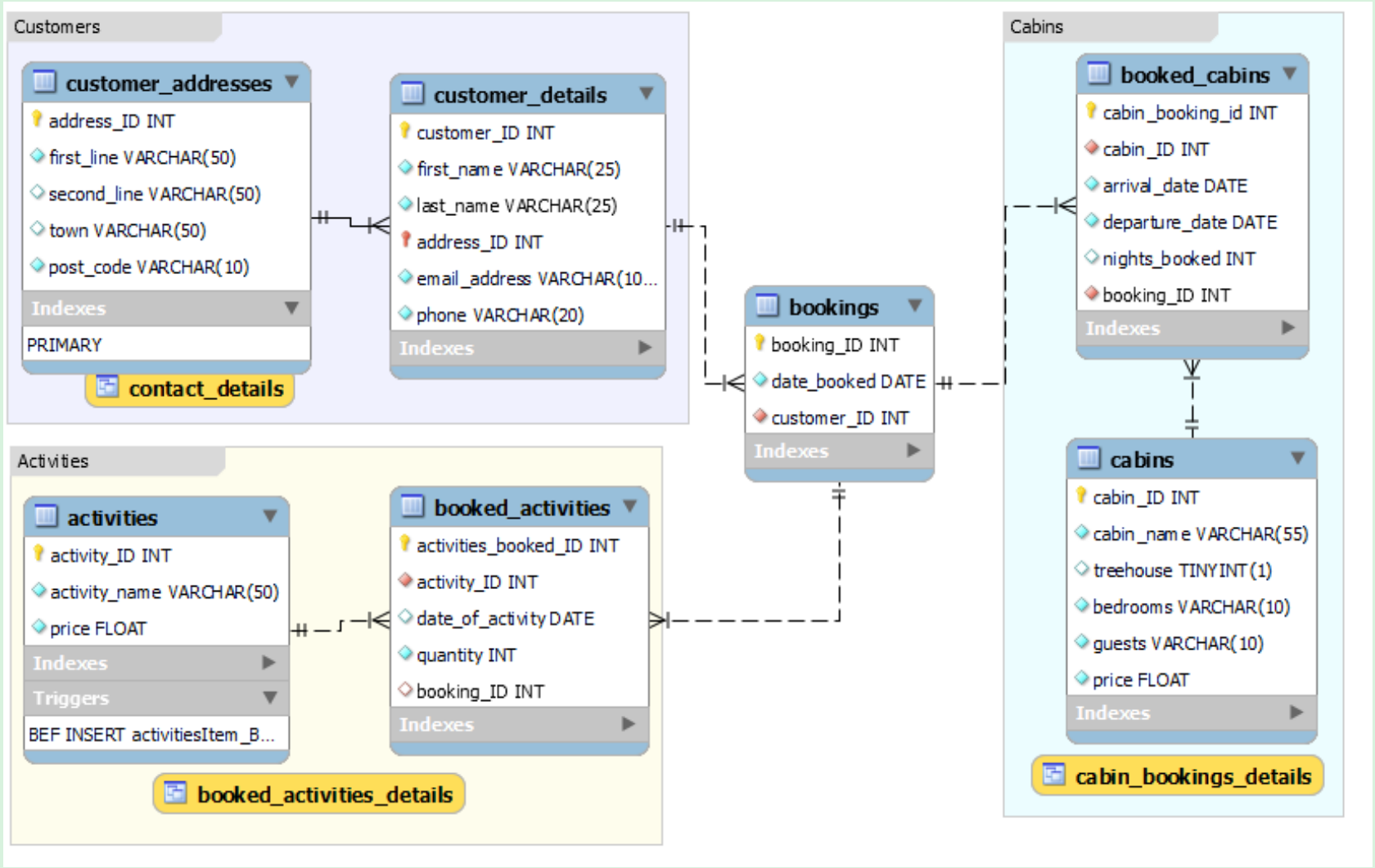
Holiday Resort

Booking system for holiday cabins and activities

Table Relations



Views

```sql
-- View to see details about activity bookings
CREATE VIEW booked_activities_details AS
    SELECT
        activities_booked_id, first_name, last_name, phone, email_address,
        activity_name, price, date_of_activity, quantity, date_booked,
        price * quantity AS total_activity_price
    FROM
        activities AS a
    INNER JOIN
        booked_activities AS ba ON ba.activity_id = a.activity_id
    INNER JOIN
        bookings AS b ON b.booking_id = ba.booking_id
    INNER JOIN
        customer_details AS cd ON cd.customer_id = b.customer_id
    ORDER BY
        activities_booked_id;

SELECT * FROM booked_activities_details;
```

```sql
300     -- Create a view for cabin details
301 •   CREATE VIEW cabin_bookings_details AS
302         SELECT
303             cabin_name, treehouse, bedrooms, guests, price, cabin_booking_id, arrival_date,
304             departure_date, nights_booked, date_booked, first_name, last_name, phone,
305             email_address, nights_booked * price AS total_price
306         FROM
307             booked_cabins AS bc
308         INNER JOIN
309             cabins AS c ON c.cabin_id = bc.cabin_id
310         INNER JOIN
311             bookings AS b ON b.booking_id = bc.booking_id
312         INNER JOIN
313             customer_details AS cd ON cd.customer_id = b.customer_id
314         ORDER BY
315             cabin_booking_id;
316
317 •   SELECT * FROM cabin_bookings_details;
```

```sql
281     -- Create a view for contact details, joining the customer details and addresses tables.
282 •   CREATE VIEW contact_details AS
283         SELECT
284             customer_id,
285             first_name,
286             last_name,
287             email_address,
288             phone,
289             first_line,
290             second_line,
291             town,
292             post_code
293         FROM
294             customer_details AS cd
295         LEFT JOIN
296             customer_addresses AS ca ON cd.address_id = ca.address_id;
297
298 •   Select * FROM contact_details;
```

Subquery–
Code and Result

```sql
-- Create a subquery to find where customers are visiting from in July
SELECT ca.town
FROM customer_addresses AS ca
WHERE address_id IN(
    SELECT cd.address_id
    FROM customer_details AS cd
    WHERE customer_ID IN(
        SELECT b.customer_ID
        FROM bookings AS b
        WHERE booking_id IN(
            SELECT bc.booking_ID
            FROM booked_cabins AS bc
            WHERE month(bc.arrival_date) = 07
            )
        )
    )
GROUP BY ca.town;
```

	town
▶	Bournemouth
	Macclesfield
	Goodwick
	Ketley
	Shrewsbury
	Northallerton

Stored Function – Code and Result

```
358    -- Create a stored function to apply a 10% discount to activities booked
359    DELIMITER //
360 •  CREATE FUNCTION discounted_price(input_amount DECIMAL(7,2))
361    RETURNS DECIMAL(7,2)
362    DETERMINISTIC
363    BEGIN
364        DECLARE output_amount DECIMAL(7,2);
365        SET output_amount = input_amount -10/100*input_amount;
366        RETURN output_amount;
367    END//
368    DELIMITER ;
369
370 •  -- Example of discount stored function
371    SELECT
372        activities_booked_id,
373        total_activity_price,
374        discounted_price(total_activity_price)AS total_after_discount
375    FROM
376        booked_activities_details;
377
```

activities_booked_id	total_activity_price	total_after_discount
1	50	45.00
2	100	90.00
3	90	81.00
4	30	27.00
5	75	67.50
6	30	27.00
7	40	36.00
8	50	45.00

Trigger

```
378     -- Create a trigger to add activities
379     DELIMITER //
380     CREATE TRIGGER activitiesItem_Before_Insert
381     BEFORE INSERT ON activities
382     FOR EACH ROW
383     BEGIN
384         SET NEW.activity_name = CONCAT(UPPER(SUBSTRING(NEW.activity_name,1,1)),
385                                 LOWER(SUBSTRING(NEW.activity_name FROM 2)));
386     END//
387     delimiter ;
388
389     INSERT INTO activities
390         (activity_ID, activity_name, price)
391     VALUES
392         (6, 'Dinner', 20.00)
```

Stored Procedure Example

```
413     -- Create a stored procedure to calculate how many surfboards are needed on a select day
414     DELIMITER //
415     CREATE PROCEDURE surfboards_needed(on_date DATE)
416     BEGIN
417         SELECT
418             SUM(quantity) AS total_surfboards_needed
419         FROM
420             Booked_activities AS ba
421         WHERE activity_id IN (
422             SELECT activity_id
423             FROM activities AS a
424             WHERE a.activity_name = 'Surfing')
425             AND ba.date_of_activity = on_date;
426     END//
427     DELIMITER ;
428
429     -- Examples:
430     CALL surfboards_needed('2023-12-13');
```

Group By Query

```sql
-- Query with group by
SELECT
cabin_name,
treehouse,
booking_id
arrival_date
FROM
    Cabins AS c
JOIN
    Booked_cabins AS bc ON bc.cabin_id = c.cabin_id
GROUP BY
    c.cabin_id,
    c.cabin_name,
    c.treehouse,
    bc.booking_id,
    bc.arrival_date
HAVING
    c.treehouse = 0
    and
    bc.arrival_date = '2023-07-09';
```