

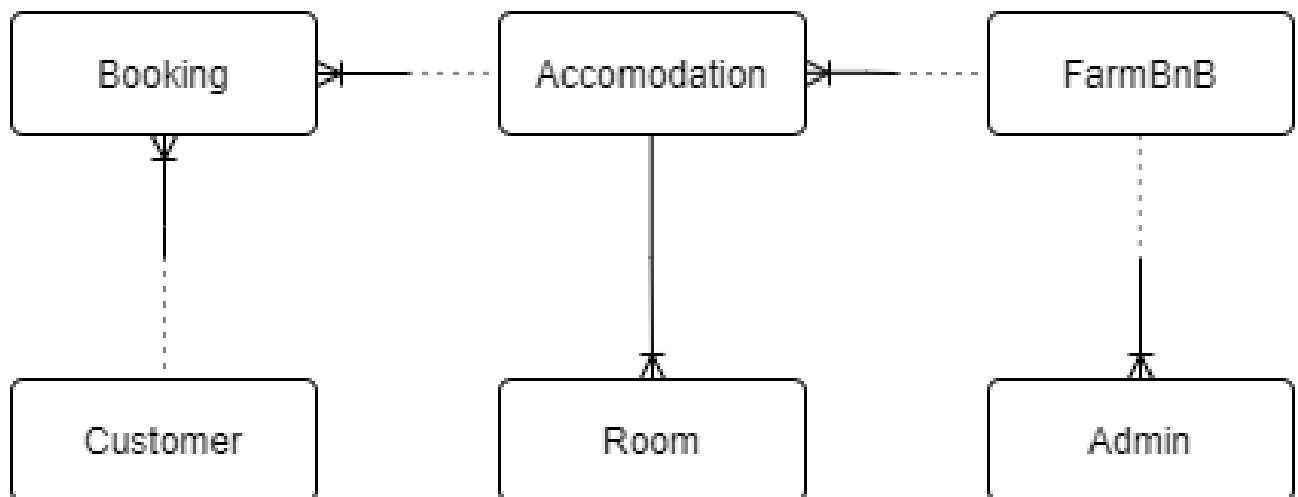
COMP2001

Assessment 1: Set Exercises

Scenario: FarmBnB

Set Exercise 1 (15 marks):

FarmBnB Entity-Relationship Diagram



Assumptions:

- Cost will be factored into Booking
- Accommodations will have multiple Bookings (database will store both history and future bookings)
- An accommodation can have multiple rooms that are listed separately
- Customer can exist without a Booking
- Accommodation can exist without any Bookings
- FarmBnB can exist without any Accomodation
- FarmBnB can exist without any Admins

Set Exercise 2 (20 marks):

Un-normalised form (UNF):

Farm Name

Farm Location

(Admin No.

Admin Name)

(Accommodation No.

Accommodation Name

Accommodation Catering

(Room No.

Room Name

Room Description

Room Status

Room Booked

(Booking No.

Arrival Date

Departure Date

Customer No.

Customer Name

Customer Address

Customer Tel No.)))

First Normal form (1NF):

Farm Name

Farm Location

Farm Name

Admin No.

Admin Name

Farm Name

Accommodation No.

Accommodation Name

Accommodation Catering

Farm Name

Accommodation No.

Room No.

Room Name

Room Description

Room Status

Room Booked

Farm Name

Accommodation No.

Room No.

Booking No.

Arrival Date

Departure Date

Customer No.

Customer Name

Customer Address

Customer Tel No.

Second Normal form (2NF):

Farm Name

Farm Location

Farm Name

Admin No.

Admin No.

Admin Name

Farm Name

Accommodation No.

Accommodation Name

Accommodation Catering

Farm Name

Accommodation No.

Room No.

Room Name

Room Description

Room Status

Room Booked

Farm Name

Accommodation No.

Room No.

Booking No.

Booking No.

Arrival Date

Departure Date

Customer No.

Customer Name

Customer Address

Customer Tel No.

Third Normal form (3NF):

Farm Name

Farm Location

Farm Name

Admin No.

Admin No.

Admin Name

Farm Name

Accommodation No.

Accommodation Name

Accommodation Catering

Farm Name

Accommodation No.

Room No.

Room Name

Room Description

Room Status

Room Booked

Booking No.

*Customer No.

*Accommodation No.

*Room No.

Arrival Date

Departure Date

Farm Name

Accommodation No.

Room No.

Booking No.

Customer No.

Customer Name

Customer Address

Customer Tel No.

Set Exercise 3 (30 marks):

CW1.AccomLookup

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE TABLE [CW1].[AccomLookup](
6      [ID] [int] IDENTITY(1,1) NOT NULL,
7      [Farm_Name] [varchar](255) NOT NULL,
8      [Room_ID] [int] NOT NULL,
9      [Accommodation_ID] [int] NOT NULL,
10     CONSTRAINT [pk_id] PRIMARY KEY CLUSTERED
11     (
12         [ID] ASC
13     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
14 ) ON [PRIMARY]
15 GO
16 ALTER TABLE [CW1].[AccomLookup] WITH CHECK ADD CONSTRAINT [fk_accom] FOREIGN KEY([Accommodation_ID])
17 REFERENCES [CW1].[Accommodation] ([Accommodation_ID])
18 GO
19 ALTER TABLE [CW1].[AccomLookup] CHECK CONSTRAINT [fk_accom]
20 GO
21 ALTER TABLE [CW1].[AccomLookup] WITH CHECK ADD CONSTRAINT [fk_farm] FOREIGN KEY([Farm_Name])
22 REFERENCES [CW1].[Farm] ([Farm_Name])
23 GO
24 ALTER TABLE [CW1].[AccomLookup] CHECK CONSTRAINT [fk_farm]
25 GO
26 ALTER TABLE [CW1].[AccomLookup] WITH CHECK ADD CONSTRAINT [fk_room] FOREIGN KEY([Room_ID])
27 REFERENCES [CW1].[Room] ([Room_ID])
28 GO
29 ALTER TABLE [CW1].[AccomLookup] CHECK CONSTRAINT [fk_room]
30 GO
```

CW1.Accommodation

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE TABLE [CW1].[Accommodation](
6      [Accommodation_ID] [int] IDENTITY(1,1) NOT NULL,
7      [Accommodation_Name] [varchar](255) NOT NULL,
8      [Accommodation_Catering] [bit] NOT NULL,
9      CONSTRAINT [pk_accommodation] PRIMARY KEY CLUSTERED
10     (
11         [Accommodation_ID] ASC
12     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
13 ) ON [PRIMARY]
14 GO
15
```

CW1.Admin

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE TABLE [CW1].[Admin](
6      [Admin_ID] [int] IDENTITY(1,1) NOT NULL,
7      [Admin_Name] [varchar](255) NOT NULL,
8      CONSTRAINT [pk_admin] PRIMARY KEY CLUSTERED
9     (
10         [Admin_ID] ASC
11     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
12 ) ON [PRIMARY]
13 GO
14
```

CW1.Booking

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE TABLE [CW1].[Booking](
6     [Accom_ID] [int] NULL,
7     [Booking_ID] [int] IDENTITY(1,1) NOT NULL,
8     [Customer_ID] [int] NULL,
9     [Arrival_Date] [date] NOT NULL,
10    [Departure_Date] [date] NOT NULL,
11    CONSTRAINT [pk_booking] PRIMARY KEY CLUSTERED
12    (
13        [Booking_ID] ASC
14    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
15    ) ON [PRIMARY]
16 GO
17 SET ANSI_NULLS ON
18 GO
19 SET QUOTED_IDENTIFIER ON
20 GO
21 CREATE TRIGGER [CW1].[MaintenanceAlert] ON [CW1].[Booking]
22 AFTER INSERT, UPDATE
23 AS
24 BEGIN
25     DECLARE @AccomRoom int;
26
27     SELECT @AccomRoom = inserted.Accom_ID
28     FROM CW1.Booking as i, inserted
29     WHERE i.Booking_ID = inserted.Booking_ID
30
31     IF ((@AccomRoom) % 10 = 0)
32     BEGIN
33
34         UPDATE CW1.Room
35         SET Room_Status = 'Maintenance Required'
36         WHERE Room_ID = @AccomRoom
37
38     END
39 END
40 GO
41 ALTER TABLE [CW1].[Booking] ENABLE TRIGGER [MaintenanceAlert]
42 GO
--
```

CW1.Customer

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE TABLE [CW1].[Customer](
6     [Customer_ID] [int] IDENTITY(1,1) NOT NULL,
7     [Customer_Name] [varchar](255) NOT NULL,
8     [Customer_Address] [varchar](255) NOT NULL,
9     [Customer_Tel] [varchar](255) NOT NULL,
10    CONSTRAINT [pk_customer] PRIMARY KEY CLUSTERED
11    (
12        [Customer_ID] ASC
13    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
14    ) ON [PRIMARY]
15 GO
16
```

CW1.Farm

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE TABLE [CW1].[Farm](
6      [Farm_Name] [varchar](255) NOT NULL,
7      [Farm_Address] [varchar](255) NOT NULL,
8      CONSTRAINT [pk_farm] PRIMARY KEY CLUSTERED
9      (
10         [Farm_Name] ASC
11     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
12 ) ON [PRIMARY]
13 GO
```

CW1.Room

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE TABLE [CW1].[Room](
6      [Room_ID] [int] IDENTITY(1,1) NOT NULL,
7      [Room_Name] [varchar](255) NOT NULL,
8      [Room_Description] [text] NULL,
9      [Room_Status] [varchar](255) NOT NULL,
10     [Room_Booked] [bit] NOT NULL,
11     CONSTRAINT [pk_room] PRIMARY KEY CLUSTERED
12     (
13         [Room_ID] ASC
14     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
15 ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
16 GO
```


Set Exercise 4 (10 marks):

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE VIEW [CW1].[BookingsView] AS
6
7  SELECT
8      CW1.Customer.Customer_ID, CW1.Customer.Customer_Name,
9      CW1.Booking.Arrival_Date, CW1.Booking.Departure_Date,
10     CW1.Accommodation.Accommodation_Name, CW1.Accommodation.Accommodation_Catering,
11     CW1.Room.Room_Name, CW1.Room.Room_Description
12
13 FROM CW1.Booking, CW1.Customer, CW1.Accommodation, CW1.Room
14 GO
--
```

Set Exercise 5 (15 marks):

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [CW1].[CreateBooking4](
6      @Accom int,
7      @Arrival date,
8      @Departure date,
9      @Customer int
10 )
11
12 AS
13 BEGIN
14     BEGIN TRY
15
16         IF EXISTS(SELECT Customer_ID FROM CW1.Customer WHERE Customer_ID = @Customer)
17             IF EXISTS (SELECT Accommodation_ID FROM CW1.Accommodation WHERE Accommodation_ID = @Accom)
18                 BEGIN
19                     INSERT INTO CW1.Booking(Accom_ID, Arrival_Date, Departure_Date, Customer_ID)
20                     VALUES(@Accom, @Arrival, @Departure, @Customer)
21                 END
22     END TRY
23     BEGIN CATCH
24         IF @@TRANCOUNT > 0
25             ROLLBACK TRANSACTION;
26     END CATCH
27 END;
28 GO
```

Set Exercise 6 (10 marks):

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE TRIGGER [CW1].[MaintenanceAlert] ON [CW1].[Booking]
6  AFTER INSERT, UPDATE
7  AS
8  BEGIN
9      DECLARE @AccomRoom int;
10
11      SELECT @AccomRoom = inserted.Accom_ID
12      FROM CW1.Booking as i, inserted
13      WHERE i.Booking_ID = inserted.Booking_ID
14
15      IF ((@AccomRoom) % 10 = 0)
16      BEGIN
17
18          UPDATE CW1.Room
19          SET Room_Status = 'Maintenance Required'
20          WHERE Room_ID = @AccomRoom
21
22      END
23  END
24  GO
25  ALTER TABLE [CW1].[Booking] ENABLE TRIGGER [MaintenanceAlert]
26  GO
27
```