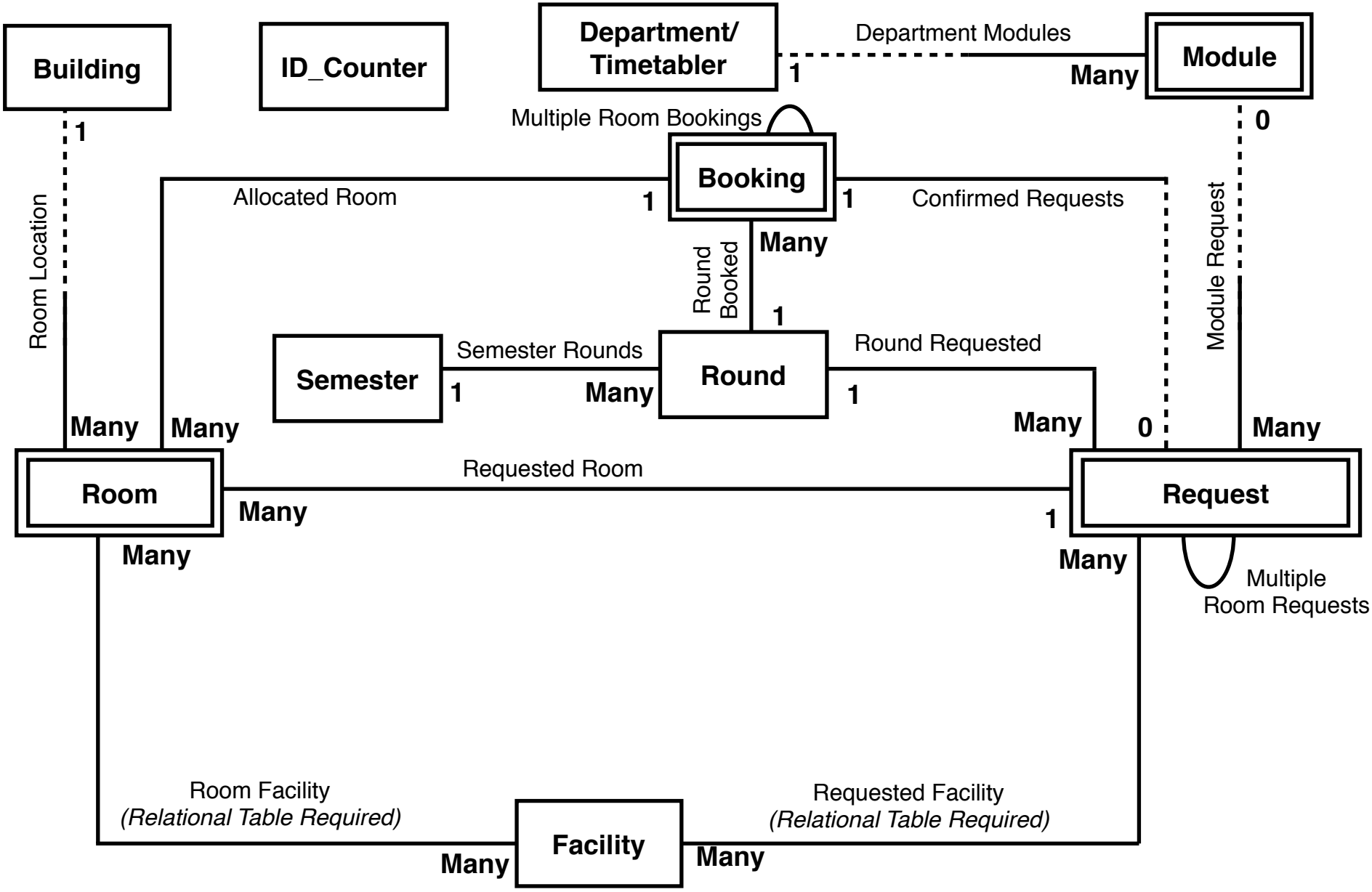
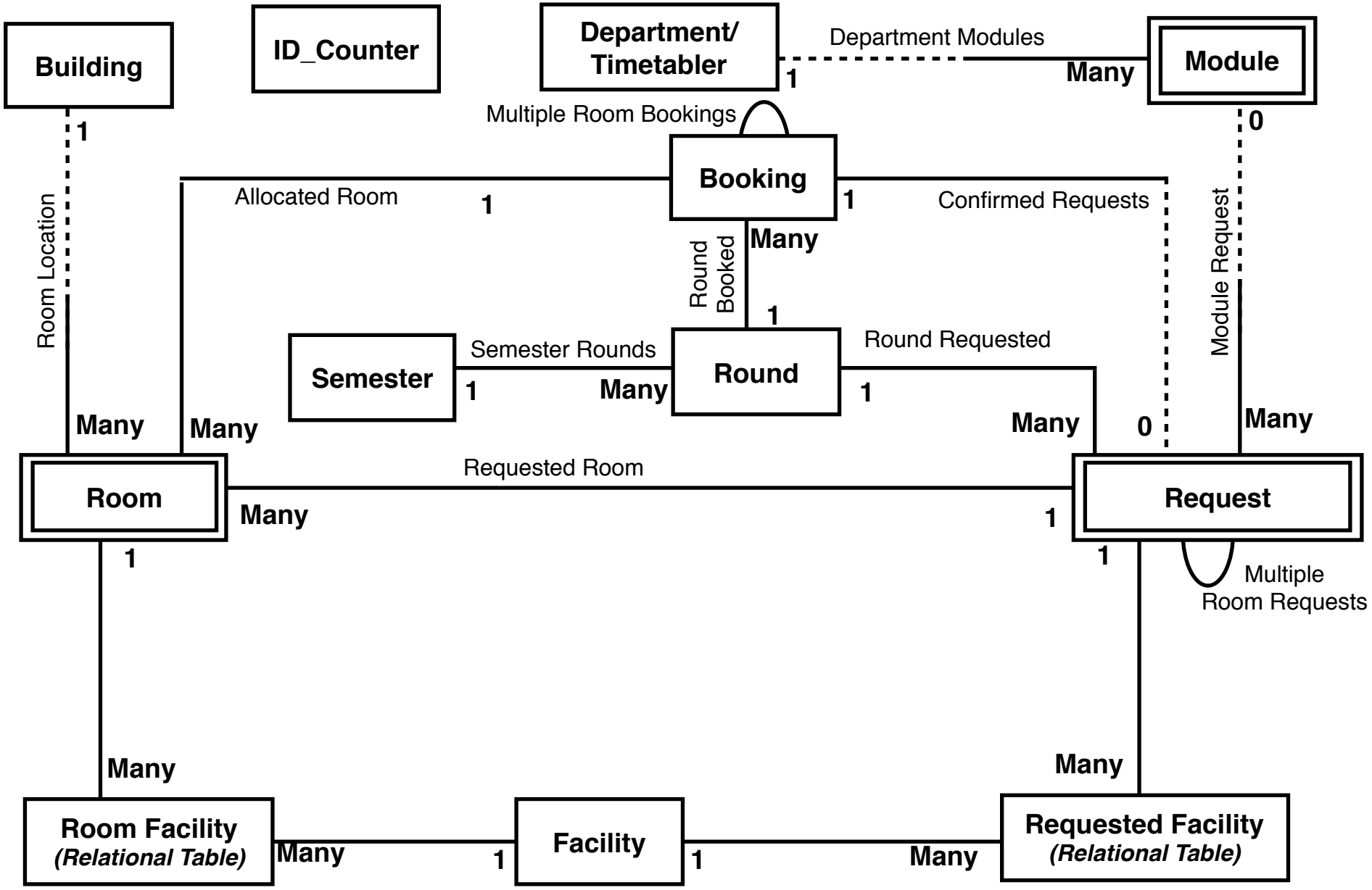
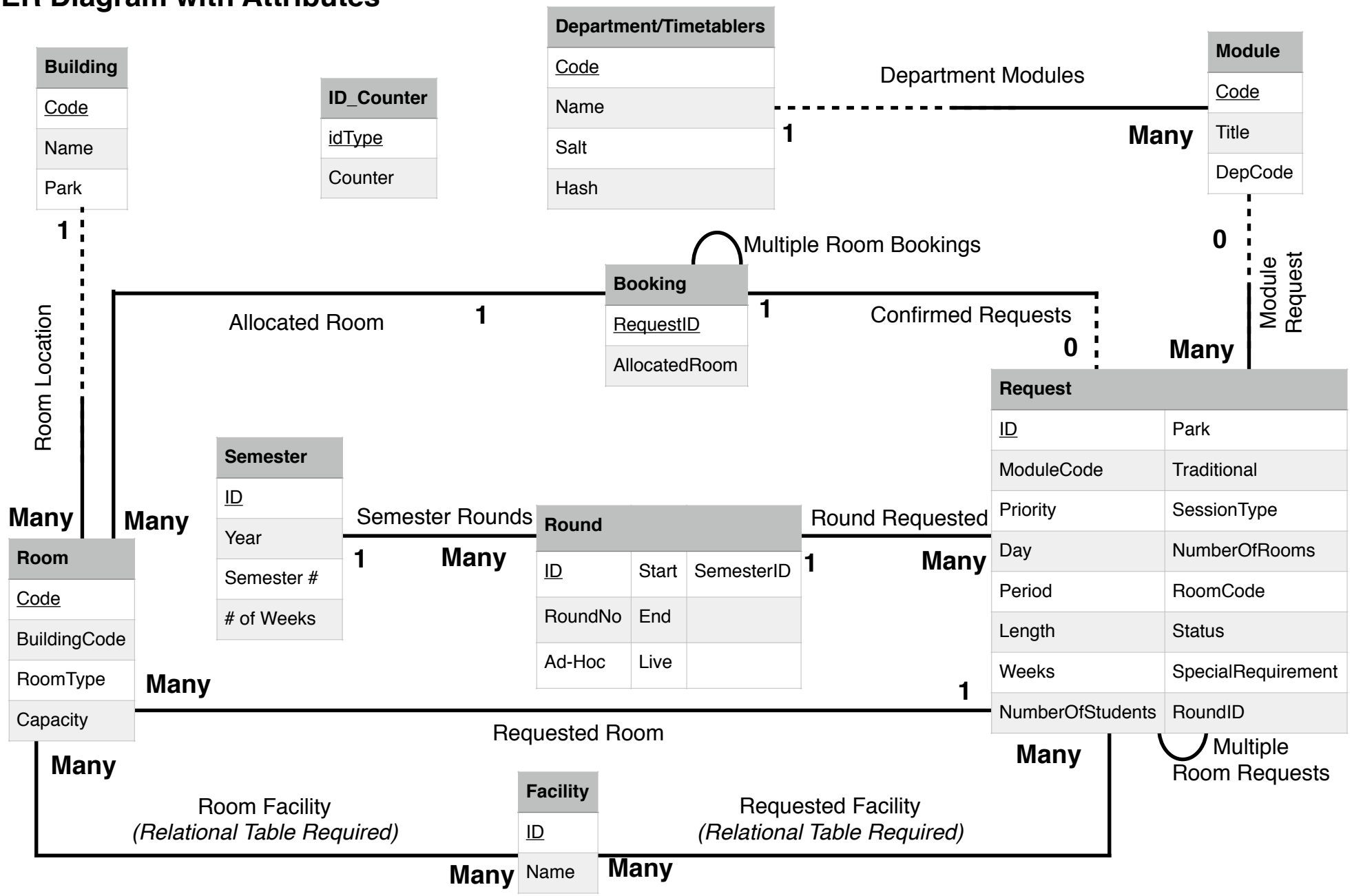
ER Diagram


Building
ID_Counter
Department/
Timetabler
Department Modules
1
Many
Module
0
Multiple Room Bookings
Booking
1
Allocated Room
1
Confirmed Requests
Many
Round
Booked
1
Room Location
1
Module Request
Semester
Semester Rounds
1
Many
Round
Round Requested
1
Many
0
Many
Many
Many
Room
Requested Room
Many
1
Request
Many
Multiple
Room Requests
Room Facility
(Relational Table Required)
Requested Facility
(Relational Table Required)
Many
Facility
Many

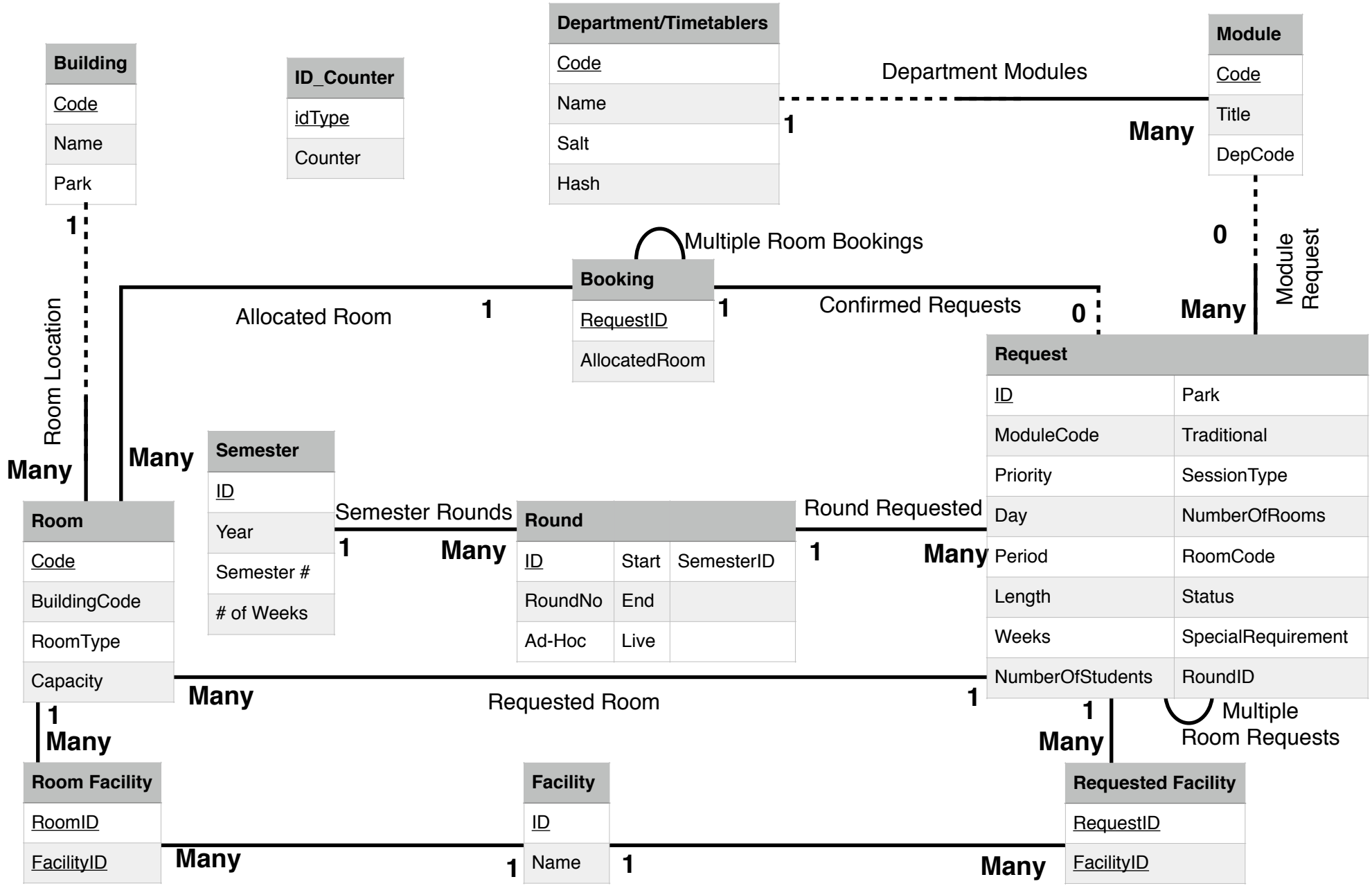
ER Diagram with Relational Tables



ER Diagram with Attributes



ER Diagram with Attributes and Relational Tables



System Database Structure

Entities

Department/Timetabler Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
Code	string	varchar(2)	Primary Key - department codes used by the University - Also the username
Name	string	varchar(255)	The full name of the Department
Salt	string	varchar(255)	String concatenated onto the password for increased security
Hash	string	varchar(255)	Password for timetabler login stored after hashing
This table contains the department's code and name, as well as the account information required for login. The table is used to manage all modules and associated requests/bookings.			

Module Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
Code	string	varchar(10)	Primary Key - module codes used by the University
Title	string	varchar(255)	The full name of the Module
DepCode	string	varchar(2)	Foreign Key from Department Table - used to identify which department module belongs to
This table contains module information, used for making requests and placing bookings.			

Facility Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
ID	int	smallint (unsigned)	Primary Key
name	string	varchar(255)	The name/title of the facility
This table stores all the facilities and available requirements for rooms and requests.			

ID_Counter Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
idType	int	smallint (unsigned)	Primary Key
Counter	string	varchar(255)	The current (highest) id value stored in the respective table
As both Request Table and Booking Table do not have a primary key as duplicates are allowed, a counter is required to be stored to manually increment the id fields for both tables.			

Building Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
Code	string	varchar(2)	Primary Key
Name	string	varchar(255)	The full name of the Building
Park	int	tinyint(1)	Used to store the location of the building: 0 - Central, 1 - East, 2 - West
This table contains the Building information, including location. This table is used to manage the records of the Room Table relatively.			

Room Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
Code	string	varchar(10)	Primary Key
BuildingCode	string	varchar(2)	Foreign Key from Building Table - used to identify which building the room is in
RoomType	int	tinyint(1)	What type the room is: 0 - Lecture, 1 - Tutorial, 2 - Labs
Capacity	Int	smallint(3) unsigned	The maximum number of students the room can accommodate
This table contains room information, including the building it is in, room type (lecture, tutorial or lab), and maximum capacity of the room. All room (and building) information required for making a request will done via this table.			

Request Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
ID	int	int (unsigned)	Primary Key
ModuleCode	string	varchar(10)	Foreign Key from Module Table - used to identify for which module the request is being made for
Priority	bool	tinyint(1)	If request is high priority, then true, otherwise false
Day	int	tinyint(1)	Which day the request is being made for 0 - Monday, 1 - Tuesday, 2 - Wednesday, 3 - Thursday, 4 - Friday
Period	int	tinyint(1)	Which period the request is being made for 0 - 09:00, 1 - 10:00, 2 - 11:00, 3 - 12:00, 4 - 13:00, 5 - 14:00, 6 - 15:00, 7 - 16:00, 8 - 17:00
Length	int	tinyint(1)	How long the request is being made for 0 - 1hr, 1 - 2hrs, 2 - 3hrs, 3 - 4hrs, 4 - 5hrs, 5 - 6hrs, 6 - 7hrs, 7 - 8hrs, 8 - 9hrs
Weeks	string	varchar(15)	A 15 digit binary format to represent weeks 1-15, where each digit represent a week, e.g. 010110011111000 = Weeks 2, ,4 - 5, 8 - 12
NumberOfStudents	int	smallint(3) unsigned	The number of student the session will have
ParkPreference	int	tinyint(1)	The park preference for the room: 0 - East, 1 - Central, 2 - West Corresponds to Park attribute in Building Table
Traditional	bool	tinyint(1)	If traditional, then true, if seminar, then false.
SessionType	int	tinyint(1)	Related to Room Table, Session type of the request: 0 - Lecture, 1 - Tutorial, 2 - Labs If traditional is false, this is automatically set to Lecture. Corresponds to RoomType attribute in Room Table
NumberOfRooms	int	tinyint (unsigned)	The number of rooms the sessions requires
RoomCode	string	varchar(10)	Foreign Key from Room Table - request's room preference. Corresponds to Code attribute from Room table
OtherRequirements	string	varchar(255)	Any other special requirements not covered by facilities
RoundID	int	int (unsigned)	Foreign Key from Round Table - the round (and time) the request is made at
Status	int	tinyint(1)	The status of the sent request: 0 - Pending, 1 - Allocated, 2 - Rejected
This table stores the majority of the request information. In order to accommodate multiple room preferences/requests for one session, the table will duplicate records with the RoomCode attribute changing for each room preference/request.			

Booking Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
RequestID	int	int (unsigned)	Primary Key - Foreign Key from Request Table
AllocatedRoom	string	varchar(10)	Foreign Key from Room Table - allocated room. Corresponds to Code attribute from Room table
This table stores all requests where Status attribute is set to “Allocated”, along with the allocated Room Code. Similar to request table, in order to accommodate bookings which require multiple rooms, records will be duplicated with the AllocatedRoom attribute changing for each room booked.			

Semester Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
ID	int	smallint (unsigned)	Primary Key
Year	string	varchar(255)	Name/Title of the facility
Semester Number	int	tinyint(1)	Semester number for that Year (group)
Number of Weeks	int	tinyint(2)	Number of Weeks in the Semester
The table stores semester data which is used to manage all the rounds in the Rounds Table.			

Round Table			
Attribute	**Programming Data Type**	**Database Data Type**	**Description**
ID	int (unsigned)	int (unsigned)	Primary Key
RoundNo	int	tinyint(1)	The current round number
Start	date/time	datetime	The start date and time of the round
End	date/time	datetime	The end date and time of the round
Ad-Hoc	bool	tinyint(1)	If current round is ad-hoc, then true, otherwise false
Live	bool	tinyint(1)	If current round is live, then true, otherwise false. Used for validation/management of requests/bookings.
SemesterID	int	smallint (unsigned)	Semester group round belongs to
This stores Round information, including start/end date/time. This also provides the system with a custom unit of time with Semester table, enabling differentiation of requests/bookings between themselves and also provides history mechanism. Only a maximum of 2 records should ever be Live at a given time, an ad-hoc round and the round for requests/bookings for next semester.			

Relationships

Name	Entity A	Entity B	Description
Department Modules	Department/ Timetabler	Module	Each department has a set of modules. However, a module cannot exist without an associative department
Module Requests	Module	Request	Each request can be made for one module. Requests cannot exist without an associative module
Room Location	Building	Room	Each room belongs to a building. A room cannot exist without an associative building
Round Requested	Round	Request	Each request is made during a particular round, ad-hoc or not. A request must have an associative round to exist
Confirmed Requests	Request	Booking	A booking is only made when a request has been allocated (confirmed/accepted)
Round Booked	Round	Booking	A booking should be made during a particular round, most likely the round the request was made
Multiple Room Requests	Request	Request	In order to accommodate multiple room requests/ preferences, duplicated records with the same RequestID are made
Multiple Room Bookings	Booking	Booking	In order to accommodate the booking of multiple rooms, duplicated records with the same RequestID are made
Request Room	Request	Room	A request can be made for a particular room
Allocated Room	Booking	Room	A booking is made for a particular room
Semester Rounds	Semester	Round	Semester manages the rounds in the Rounds table. Both Semester and Rounds table form the custom unit of time for the system
Requested Facility	Request	Facility	**Relational Table Required** A request can be made with some required facilities. The relational table consists of the IDs from each table.
Room Facility	Room	Facility	**Relational Table Required** A room can accommodate many facilities. The relational table consists of the IDs from each table.