TABLE OF CONTENTS

CONTENTS

1. INTRODUCTION
2. EXECUTIVE SUMMARY
3. METHODOLOGY
4. DATA ARCHITECTURE
5. DATA MODEL
6. ENTITY SPECIFICATION FORM
7. DATA DEFINITION LANGUAGE
8. SQL QUERIES
9. CONCLUSION

Final Report:

Scheduling Application

1. INTRODUCTION:

This project aims to design and develop a web-based scheduling application for a coworking space company to manage and schedule its resources, including conference rooms, LCD projectors, and portable PCs. The application will provide real-time availability of resources, avoid conflicting bookings, generate reports on resource utilization, and identify the most frequent customers utilizing the services.

i. GOAL:

- The goal of this project is to develop scheduling applications that will enable users to replace their current paper-based or simple Word document scheduling systems with a database management system. This will help them benefit from efficient data collection and information sharing. The new system will include an online scheduling page accessible to both clients and users, with an intuitive user interface for easy navigation and seamless scheduling. Users can schedule and book various resources such as conference halls, small conference rooms, LCD projectors, and portable PCs for specific dates and periods.

- Additionally, the system will be able to generate reports on resource utilization per week, month, and year. It will also be able to answer queries such as when a specific resource will be available between certain hours, the average weekly occupancy of a conference room, and identify the most frequent customers.

ii. BACKGROUND:

Our Coworking space company relies on the scheduling application to streamline and optimize resource management. Our company provides a range of amenities to its clients, including a Conference Hall, 10 Small conference rooms, 7 LCD projectors, and 5 Portable PCs, with additional resources slated for availability shortly. These resources are leased out to customers regularly, and management requires weekly, monthly, and yearly reports on their utilization. The scheduling application offers valuable insights into room occupancy, resource usage, and customer activity, providing information as needed to ensure smooth operations.

2. EXECUTIVE SUMMARY:

i. DESCRIPTION:

The current process for scheduling and tracking resources at the coworking space company is handled manually. However, with the company's steady expansion, managing the growing number of customers and resources in this manner has become increasingly challenging. As a result, the company requires a centralized system to streamline the scheduling process effectively. Such a system will also prevent inadvertent double-booking of resources by different users, among other advantages.

ii. PROBLEM:

In our modern world, keeping track of appointments is crucial for both personal and professional obligations. Fortunately, scheduling applications and electronic calendars have made this task much easier. However, our coworking space company currently relies on paper-based or basic Word documents to manage appointment data, which presents a variety of challenges:

- Repetitive data
- Inaccuracy
- Limited adaptability
- Weak security
- Limited data sharing and access

To address these issues, it is essential to adopt a more efficient and secure system for managing appointment data.

3. METHODOLOGY:

Our proposed approach involves leveraging the capabilities of the MySQL database management system. This solution is specifically designed to handle and manage data efficiently and is widely adopted in organizations to streamline data collection and facilitate seamless integration. This approach guarantees swift and convenient access to information for users while preventing any unnecessary data duplication.

i. INFORMATION NEEDS IDENTIFICATION:

- ❖ SIMPLE DATA MANAGEMENT

  By centralizing all data in one location, there is no need for duplicate entries. This allows every user to access information directly without having to reproduce it, as is often necessary with paper-based systems. Additionally, database management systems can reduce costs associated with duplicating data that results from using single application programs.

- ❖ ACCURACY

  When data is saved on multiple computers through database management software, it becomes easier to manage. Organizations use these systems to ensure the accuracy of information and maintain high levels of quality.

- ❖ DATA SECURITY

  Authorized personnel are the only ones who can access data, which prevents any unauthorized access by malicious individuals.

- ❖ IMPROVED PROCESSING SPEED

Functions such as calculations, sums, and counts are easily automated. Data management systems organize applications for reporting, storing, and logically updating data.

- REUSABLE DATA CODE

Inventory data, for example, is stored separately from other sets of data. However, queries can be combined from all modules without requiring changes to the structure. Since all data is stored in a single database, loss of information may occur in the event of a power outage or a corrupted database.

ii. **FUNCTIONS:**

The Database Management System performs the following functions.

- Data Dictionary Management
- Data Storage Management
- Data Transformation and Presentation
- Security Management
- Multi-User Access Control
- Backup and Recovery Management
- Data Integrity Management
- Database Access Languages and Application Interface
- Database Communication Interface

The Database Management System (DBMS) is responsible for managing and organizing data in a database. It performs various functions such as data dictionary management, data storage management, data transformation and presentation, security management, multiuser access control, backup and recovery management, data integrity management, database access languages and application interface, and database communication interface.

Data dictionary management involves storing metadata, which is a collection of definitions of data elements and their relationships. Metadata includes the definition of data, data types, relationships between data, integrity constraints, and more. Changes made to the database structure are automatically reflected in the data dictionary, removing structural and data dependency from the system.

Data storage management involves creating complex structures to store data. Users are freed from defining, programming, and implementing the complex physical data characteristics.

(Data Transformation and Presentation)

The DBMS supports data independence, translating logical requests into commands that physically locate and retrieve the requested data. The DBMS formats the physically retrieved data according to the logical data format specifications.

Security management involves creating a security system that enforces user security and data privacy within the database. Security rules determine the access rights of users and read/write access is granted to users as specified using access rights.

Multiuser access control ensures that multiple users can access the database concurrently without compromising the integrity of the database. The database ensures data integrity and data consistency.

Backup and recovery management provides backup and data recovery procedures to ensure data safety and integrity. DBMS systems provide special utilities that allow the Database Administrator (DBA) to perform routine and special backup and restore procedures. Recovery management deals with the recovery of the database after a failure.

Data integrity management promotes and enforces integrity rules to eliminate data integrity problems, minimizing data redundancy and maximizing data consistency.

The DBMS provides data access via a non-procedural query language. Users only need to specify what must be done without specifying how it is to be done. The DBMS's query language contains two components: a data definition language (DDL) and a data manipulation language (DML). The DBMS also provides data access to programmers via programming languages.

(Database Communication Interface)

Different users may access the database through a computer network environment. Thus, the DBMS provides communication functions to access the database through a computer network environment.

4. DATA ARCHITECTURE:

i. Initial list of Entities:

An entity is an object that exists and is distinguishable from other objects. An entity may be concrete (a person or a book, for example) or abstract. An entity is represented by a set of attributes. Entities to be used in our project of scheduling applications:

1. Customer
2. ResourceA
3. ResourceB
4. Reservation
5. ResourceUtilization
6. Availability
7. UsuageReport
8. CustomerUsuage
9. ConferenceHall
10. SmallConferenceRoom
11. LCDProjector
12. PortablePC

ii. **ATTRIBUTES:**

An attribute is a function that maps an entity set into a domain. One can characterize it as an “Attribute management system” where attributes are small chunks of information that describe something. Attributes to be used against each entity in our project area:

1. Customer:

- CustomerID (primary key)
- CustomerName
- CustomerGender
- ContactNumber
- Address
- SigninDate

2. ResourceA:

- HallID (Primary Key)
- HallName
- HallCapacity
- Location
- HallDescription
- HallPricePerHour

3. ResourceB:

- ItemID (Primary Key)
- ItemName
- ItemCapacity
- ItemDescription
- ItemPricePerHour

4. Reservation:

- ReservationID (Primary Key)
- HallID (Foreign Key referencing HallID in ResourceA)
- CustomerID (Foreign Key referencing CustomerID in Customer)
- ReservationDate
- StartTime
- EndTime

5. ResourceUtilization:

- UtilizationID (Primary Key)
- HallID (Foreign Key referencing HallID in ResourceA)
- UtilizationDate
- StartTime
- EndTime

6. Availability:

- AvailabilityID (Primary Key)
- HallID (Foreign Key referencing HallID in ResourceA)
- AvailabilityDate
- StartTime
- EndTime
- IsAvailable

7. UsageReport

- ReportID (Primary Key)
- HallID (Foreign Key referencing HallID in ResourceA)
- ReportDate
- UtilizationDuration

8. CustomerUsage:

- UsageID (Primary Key)
- CustomerID (Foreign Key referencing CustomerID in Customer)
- ResourceID (Foreign key referencing HallID in ResourceA)
- UsuageDate
- StartTime
- EndTime

These attributes provide specific information for each entity in the context of the coworking space scenario. They include unique identifiers (Primary Keys) for each table, as well as other relevant details such as names, capabilities, addresses, dates, times, and durations.

With these attributes, we will perform queries such as (example):

- Retrieve the next available day for specific resources between 1:00 and 5:00.

iii. **Initial list of Entities (Tables)**

Customer:

Field	**Description**	**Type**	**Length**
ID	Customer ID	Int	8
Name	Customer Name	Varchar	255
Gender	Customer Gender	Varchar	1
Number	Contact Number	Int	8
Address	Customer Address	Varchar	100
SigninDate	Customer Sign in Date	Date	

ResourceA:

Field	Description	Type	Length
ID	Hall ID	Int	8
NAME	Hall Name	Varchar	255
Capacity	Hall Capacity	Int	8
Location	Hall Location	Varchar	100
Description	Hall Description	Varchar	255
Price	Hall Price Per Hour	Int	8

ResourceB:

Field	Description	Type	Length
ID	Item ID	Int	8
NAME	Item Name	Varchar	255
Capacity	Item Capacity	Int	8
Description	Item Description	Varchar	255
Price	Item Price Per Hour	Int	8

Reservation:

Field	Description	Type	Length
ID	Reservation ID	Int	8
HID	Hall ID	Int	8
CID	Customer ID	Int	8
RDATE	Reservation Date	Date	
STIME	Start Time	Time	8
ETIME	End Time	Time	8

Resource Utilization:

Field	Description	Type	Length
ID	Utilization ID	Int	8
HID	Hall ID	Int	8
UDATE	Utilization DATE	Date	
USTIME	Utilization Start Time	Time	8
UETIME	Utilization End Time	Time	8

Availability:

Field	Description	Type	Length
ID	Availability ID	Int	8
HID	Hall ID	Int	8
DATE	Availability DATE	Date	
STIME	Start Time	Time	8
ETIME	End Time	Time	8
Available	Is Available	Varchar	255

Usage Report:

Field	Description	Type	Length
ID	Report ID	Int	8
HID	Hall ID	Int	8
DATE	Report Date	Date	
Duration	Utilization Duration	Time	8

Customer Usage:

Field	Description	Type	Length
ID	Usage ID	Int	8
CID	Customer ID	Int	8
RID	Resource ID	Int	8
DATE	Usage Date	Date	
STIME	Start Time	Time	8
ETIME	End Time	Time	8

5. DATA MODEL:

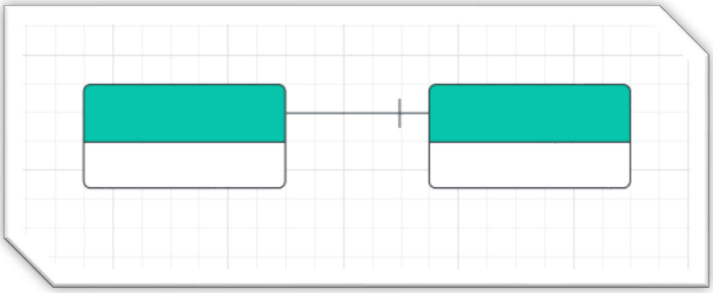
Entity Relationship Model using UML Notation

An entity-relationship model, or ER diagram, is a visual representation of entities and their relationships with each other. ER models are commonly used in computing to organize data within databases or information systems. An entity is a piece of data that represents an object or concept, and data is stored about it. A relationship represents how the data is shared between entities.

There are three types of relationships between entities:

- ❖ ONE-TO-ONE:

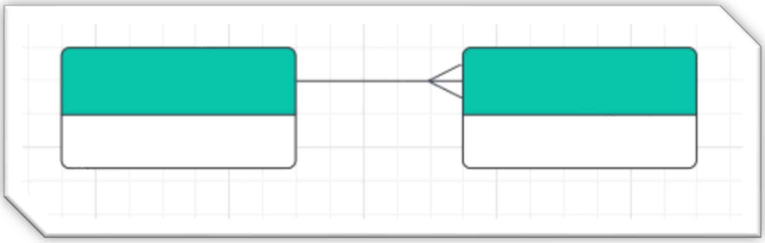
A One-to-One relationship occurs when one instance of an entity is associated with only one instance of another entity. For instance, a roof can only cover one building, and a building can only be covered by one roof. In a diagram, this type of relationship is represented by a line connecting the two entities.



- ONE-TO-MANY:

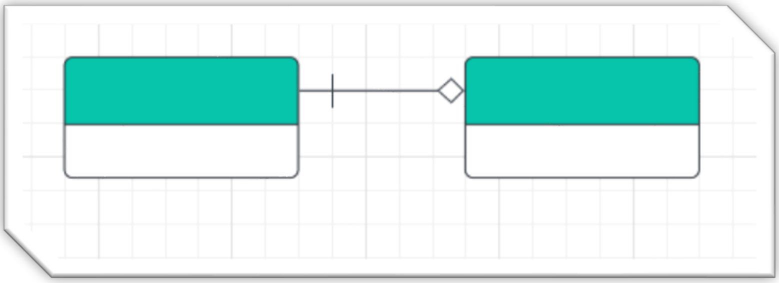
A One-to-Many relationship occurs when one instance of an entity is linked to multiple occurrences of another entity. For instance, a department may have many employees, and each employee works in only one department.

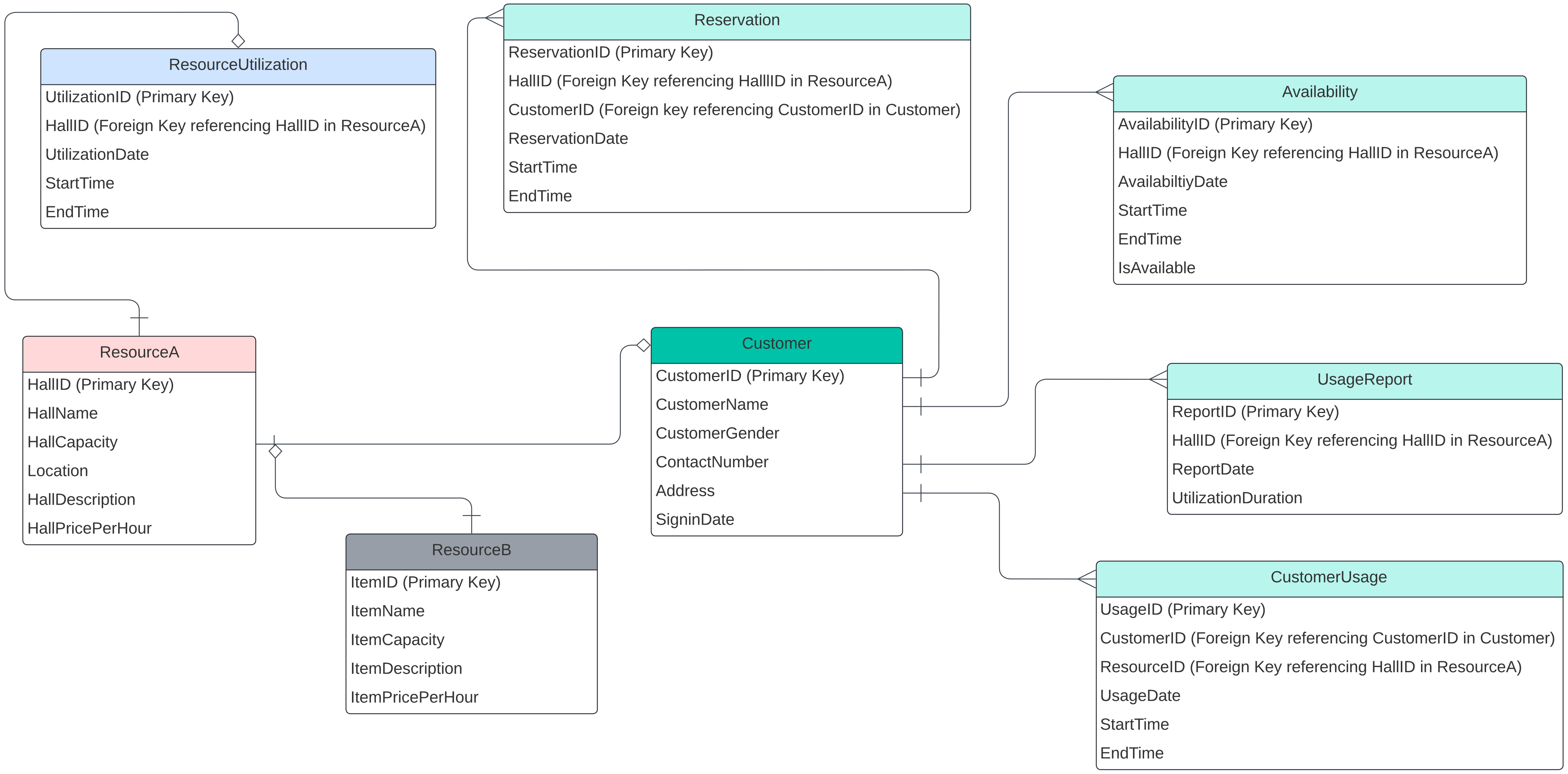
To represent a One-to-Many relationship in a diagram, you can use a line connecting the two entities, with a crow's foot symbol at the 'many' ends of the relationship.



- MANY-TO-MANY:

This type of relationship occurs when multiple entities of one type are related to multiple entities of another type.





ResourceUtilization

UtilizationID (Primary Key)
HallID (Foreign Key referencing HallID in ResourceA)
UtilizationDate
StartTime
EndTime

Reservation

ReservationID (Primary Key)
HallID (Foreign Key referencing HalllID in ResourceA)
CustomerID (Foreign key referencing CustomerID in Customer)
ReservationDate
StartTime
EndTime

Availability

AvailabilityID (Primary Key)
HallID (Foreign Key referencing HallID in ResourceA)
AvailabiltiyDate
StartTime
EndTime
IsAvailable

ResourceA

HallID (Primary Key)
HallName
HallCapacity
Location
HallDescription
HallPricePerHour

Customer

CustomerID (Primary Key)
CustomerName
CustomerGender
ContactNumber
Address
SigninDate

UsageReport

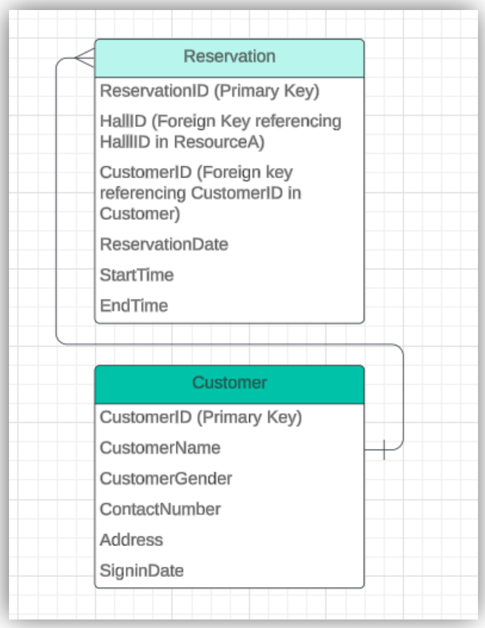
ReportID (Primary Key)
HallID (Foreign Key referencing HallID in ResourceA)
ReportDate
UtilizationDuration

ResourceB

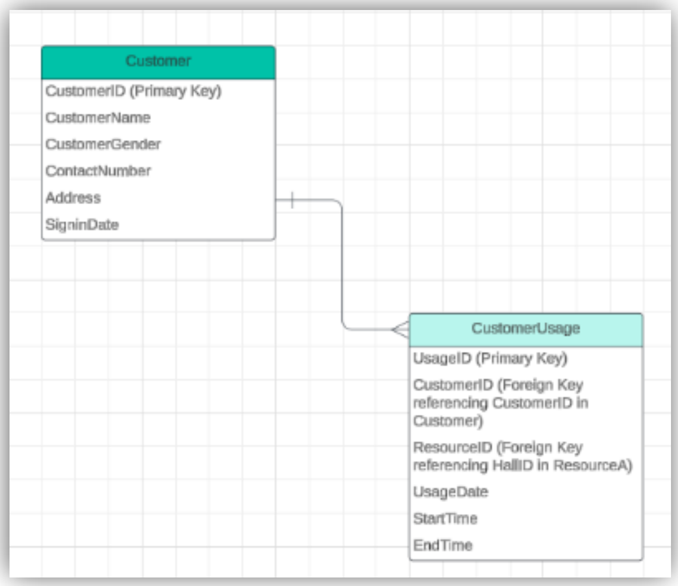
ItemID (Primary Key)
ItemName
ItemCapacity
ItemDescription
ItemPricePerHour

CustomerUsage

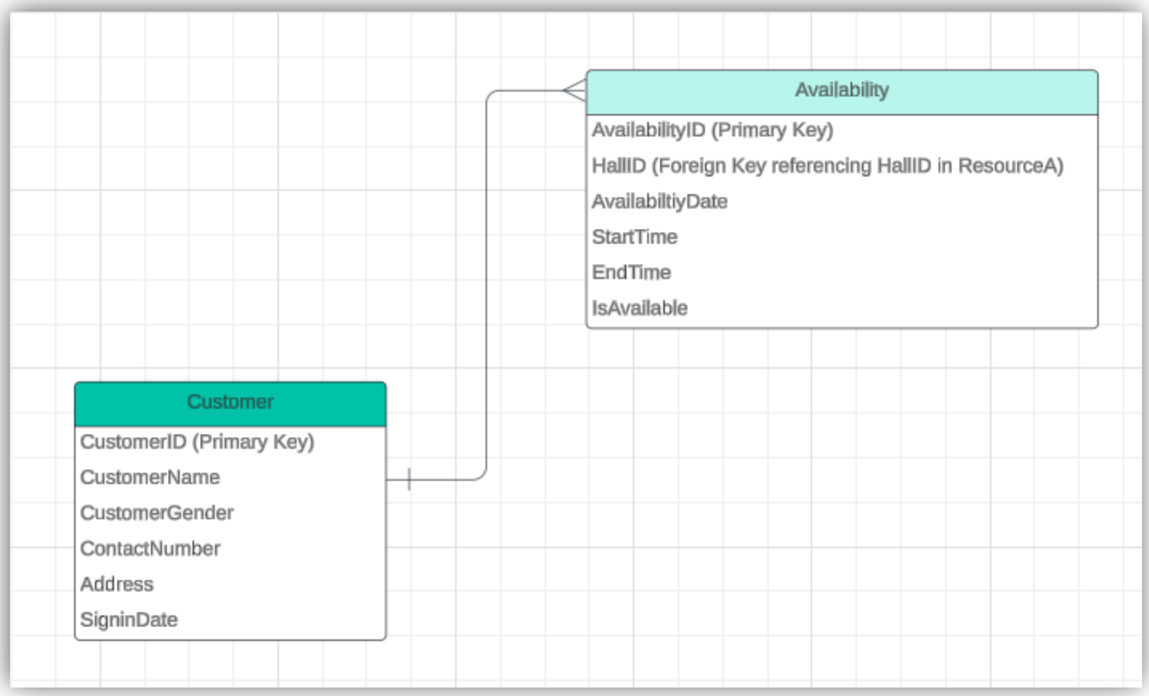
UsageID (Primary Key)
CustomerID (Foreign Key referencing CustomerID in Customer)
ResourceID (Foreign Key referencing HallID in ResourceA)
UsageDate
StartTime
EndTime


Reservation
ReservationID (Primary Key)
HallID (Foreign Key referencing HallIID in ResourceA)
CustomerID (Foreign key referencing CustomerID in Customer)
ReservationDate
StartTime
EndTime
Customer
CustomerID (Primary Key)
CustomerName
CustomerGender
ContactNumber
Address
SigninDate

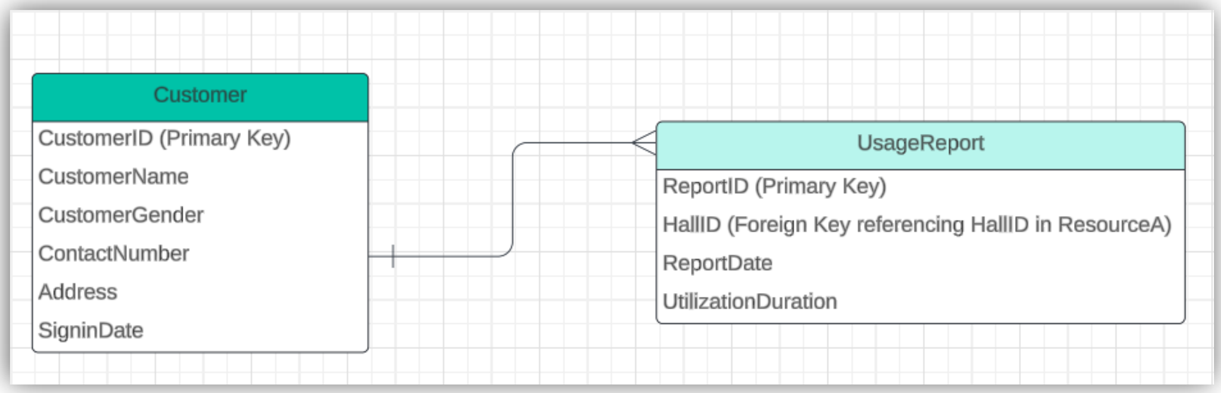
- ➢ Each Customer may have made zero or more Reservations.
- ➢ One Reservation must be made by one and only one Customer.



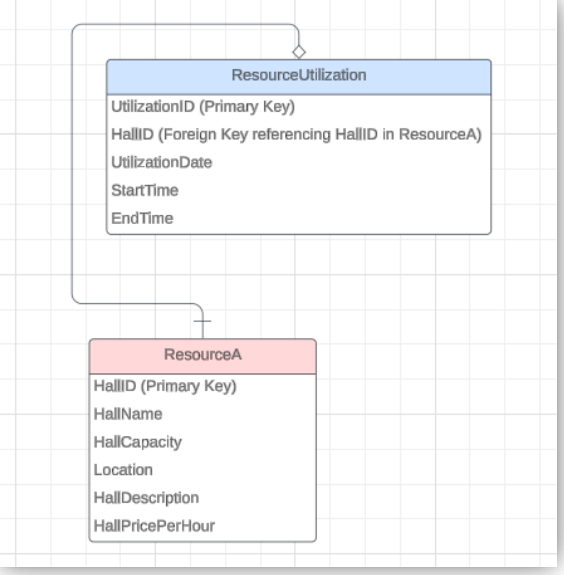
- ➢ Each Customer may have 1 or more CustomerUsage information.
- ➢ One CustomerUsage must belong to one and only one Customer.



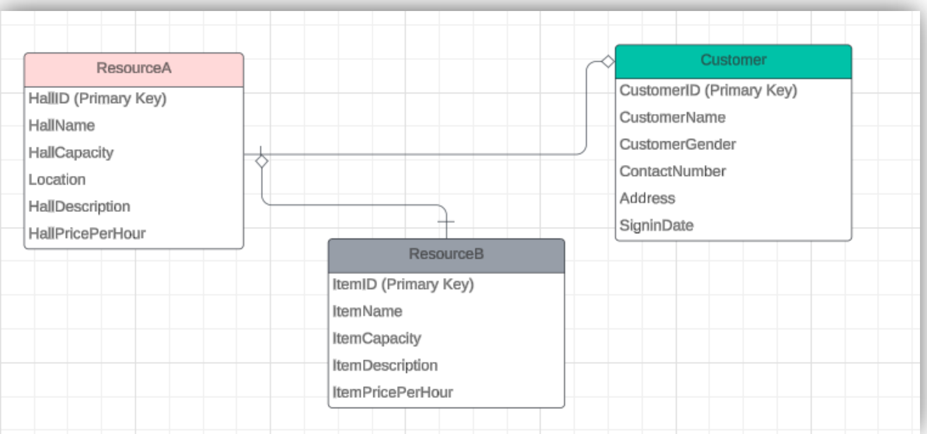
- Each Customer can check the Availability of the Resources they want to utilize.
- One Availability must be linked to one and only one Hall.
- One Availability may occur on one and only one date.
- One Availability may have a specific start time and end time.
- One Availability can be marked as available and unavailable (IsAvailable).



- One UsageReport may contain information about one and only one Hall.
- One UsageReport may have a specific report date.

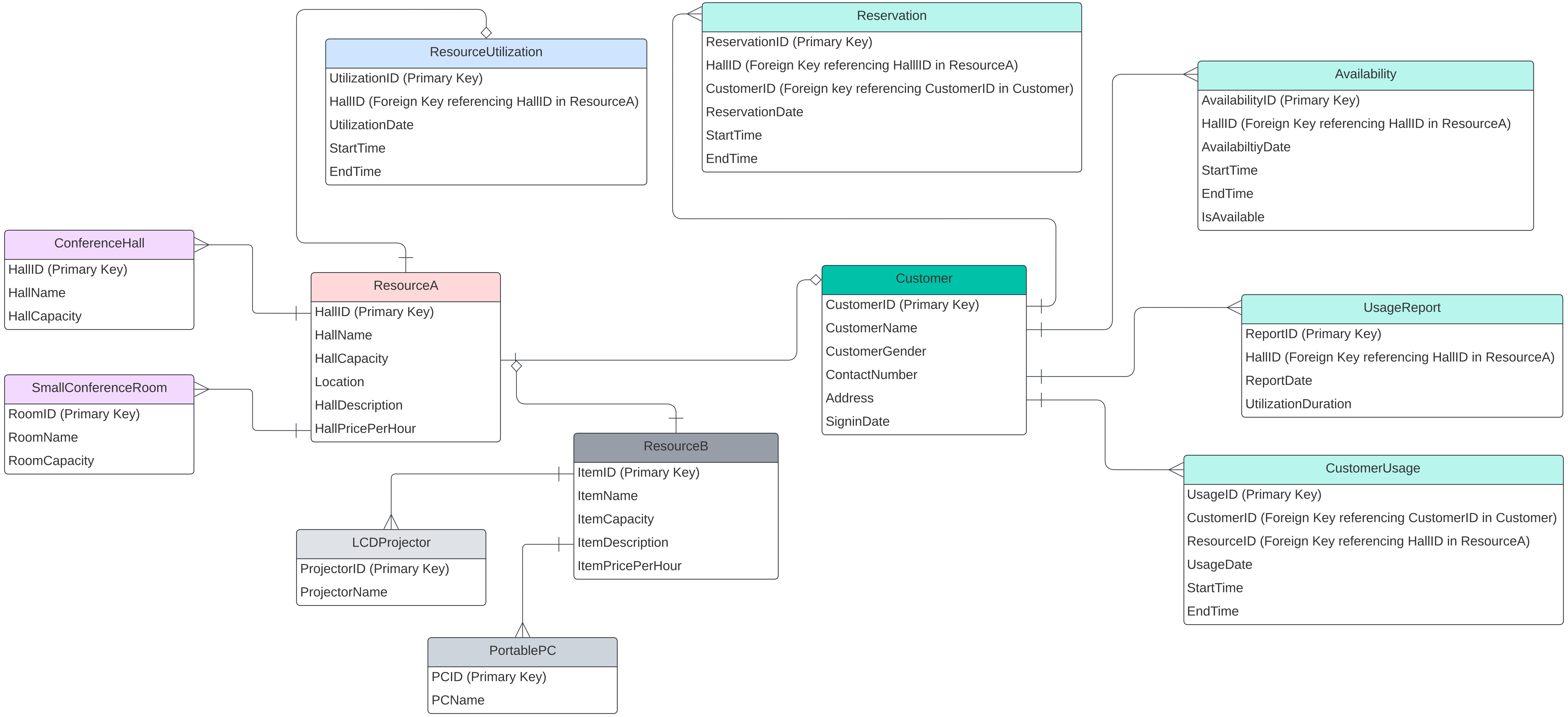


- ➢ One Hall may have zero or more ResourceUtilizations.
- ➢ One ResourceUtilizations must be associated with one and only one Hall.
- ➢ One Hall may be included in Zero or more Reservations.
- ➢ One Reservation must include one and only one Hall.
- ➢ One Hall may have zero or more Availabilities.
- ➢ One Availability must be linked to one and only one Hall.
- ➢ One Hall may appear in zero or more UsageReports.
- ➢ One Hall may have zero or more CustomerUsages.



- ➢ One item may be included in zero or more Reservations.
- ➢ One Reservation must include one and only one item.

FULL IMAGE;



ResourceUtilization
- UtilizationID (Primary Key)
- HallID (Foreign Key referencing HallID in ResourceA)
- UtilizationDate
- StartTime
- EndTime

Reservation
- ReservationID (Primary Key)
- HallID (Foreign Key referencing HalllID in ResourceA)
- CustomerID (Foreign key referencing CustomerID in Customer)
- ReservationDate
- StartTime
- EndTime

Availability
- AvailabilityID (Primary Key)
- HallID (Foreign Key referencing HallID in ResourceA)
- AvailabiltiyDate
- StartTime
- EndTime
- IsAvailable

ConferenceHall
- HallID (Primary Key)
- HallName
- HallCapacity

ResourceA
- HallID (Primary Key)
- HallName
- HallCapacity
- Location
- HallDescription
- HallPricePerHour

Customer
- CustomerID (Primary Key)
- CustomerName
- CustomerGender
- ContactNumber
- Address
- SigninDate

UsageReport
- ReportID (Primary Key)
- HallID (Foreign Key referencing HallID in ResourceA)
- ReportDate
- UtilizationDuration

SmallConferenceRoom
- RoomID (Primary Key)
- RoomName
- RoomCapacity

ResourceB
- ItemID (Primary Key)
- ItemName
- ItemCapacity
- ItemDescription
- ItemPricePerHour

CustomerUsage
- UsageID (Primary Key)
- CustomerID (Foreign Key referencing CustomerID in Customer)
- ResourceID (Foreign Key referencing HallID in ResourceA)
- UsageDate
- StartTime
- EndTime

LCDProjector
- ProjectorID (Primary Key)
- ProjectorName

PortablePC
- PCID (Primary Key)
- PCName

6. ENTITY SPECIFICATION FORM:

NORMALIZATION:

Database normalization is a technique used to organize the data in a relational database. The main aim of this technique is to minimize redundancy and dependency in the database structure. In this process, large tables are divided into smaller tables, and relationships are defined between them. This isolation of data ensures that any addition, deletion, or modification of data can be made in one table and then be propagated through the rest of the database via established relationships. By doing so, it helps in maintaining consistency and avoiding data duplication.

When creating a database from an entity-relationship model, the primary issue that arises is redundancy. Redundancy refers to the practice of storing the same data in multiple locations, which can lead to several problems. These problems include:

- Extra storage space: storing the same data in many places takes a large amount of disk space.
- Entering the same data more than once during data insertion.
- Deleting data from more than one place during deletion.
- Modifying data in more than one place.
- Anomalies may occur in the database if insertion, deletion, modification, etc are not done properly. It creates inconsistency and unreliability in the database.

To address certain issues with a database that is designed based on the ER model, a process of refinement called normalization is required. This process involves removing various types of redundancy and anomalies through a step-by-step approach that follows specific rules at each step. By doing so, the database becomes more streamlined and freer of impurities, resulting in a cleaner look that is easier to work with.

Normalization is a process that helps to eliminate certain issues that can occur in a relational database. These issues include duplication of data, insert anomaly, delete anomaly, and update anomaly.

- Duplication of data refers to the same data being listed in multiple lines of the database, which can cause confusion and inconsistencies. Insert anomaly occurs when a record about an entity cannot be inserted into the table without first inserting information about another entity. For example, you cannot enter a customer without a sale order.

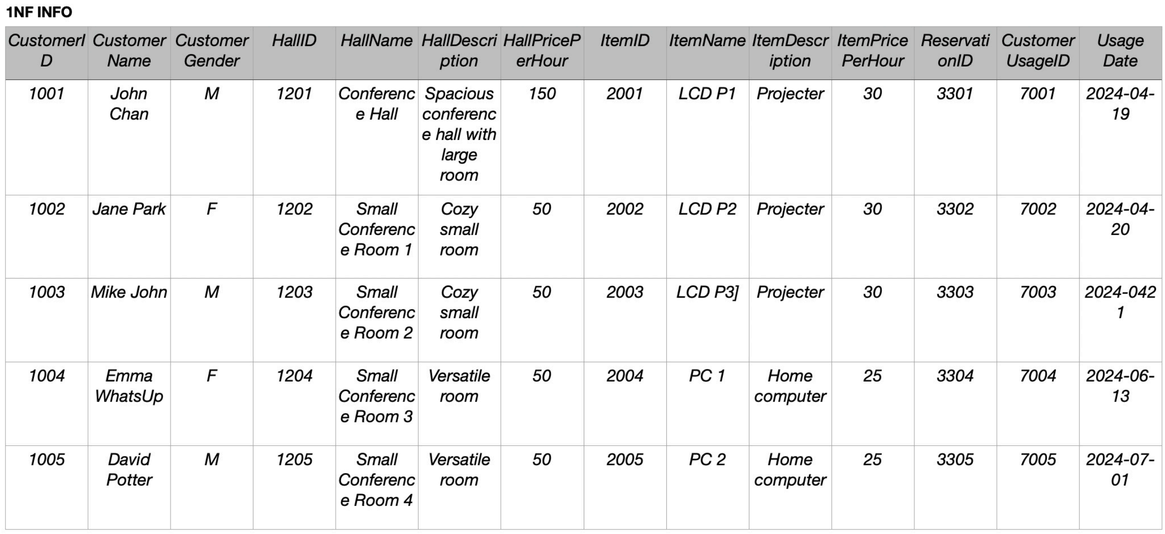
- Delete anomaly occurs when a record cannot be deleted without deleting a record about a related entity. For instance, you cannot delete a sales order without deleting all the customer's information.

- Update anomaly refers to the inability to update information without changing information in many places. To update customer information, it must be updated for each sales order the customer has placed.

Normalization helps to eliminate these issues, making the database more organized, efficient, and easier to manage.

Normalisation is a process of organising data in a database to eliminate redundancy and improve data integrity. By applying these normalisation forms, we can design a database schema that minimise data duplication and ensures efficient data storage and retrieval.

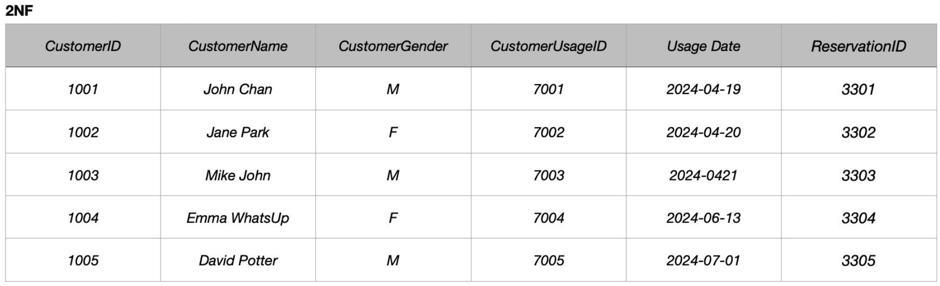
1NF (First Normal Form) is a database normalisation form that ensures that each column in a table contains only atomic values (indivisible values) and there are no repeating groups of columns. In other words, each cell in a table should hold a single value, and there should be no duplicate rows.



1NF INFO

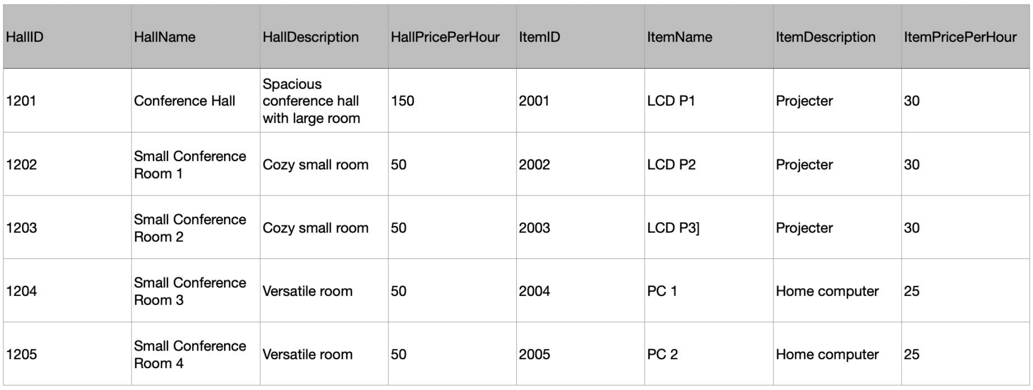
CustomerID	Customer Name	Customer Gender	HallID	HallName	HallDescription	HallPricePerHour	ItemID	ItemName	ItemDescription	ItemPrice PerHour	ReservationID	Customer UsageID	Usage Date
1001	John Chan	M	1201	Conference Hall	Spacious conference hall with large room	150	2001	LCD P1	Projecter	30	3301	7001	2024-04-19
1002	Jane Park	F	1202	Small Conference Room 1	Cozy small room	50	2002	LCD P2	Projecter	30	3302	7002	2024-04-20
1003	Mike John	M	1203	Small Conference Room 2	Cozy small room	50	2003	LCD P3]	Projecter	30	3303	7003	2024-0421
1004	Emma WhatsUp	F	1204	Small Conference Room 3	Versatile room	50	2004	PC 1	Home computer	25	3304	7004	2024-06-13
1005	David Potter	M	1205	Small Conference Room 4	Versatile room	50	2005	PC 2	Home computer	25	3305	7005	2024-07-01

2NF (Second Normal Form) is a database normalization form that builds upon 1NF and ensures that each non-key attribute (column) in a table is fully dependent on the entire primary key. It eliminates partial dependencies, which occur when a non-key attribute depends on only a part of the primary key.



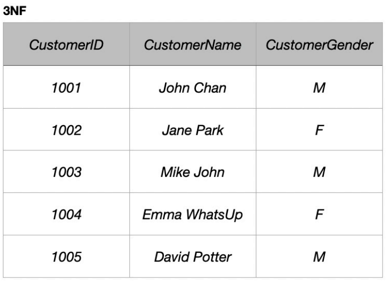
2NF

CustomerID	CustomerName	CustomerGender	CustomerUsageID	Usage Date	ReservationID
1001	John Chan	M	7001	2024-04-19	3301
1002	Jane Park	F	7002	2024-04-20	3302
1003	Mike John	M	7003	2024-0421	3303
1004	Emma WhatsUp	F	7004	2024-06-13	3304
1005	David Potter	M	7005	2024-07-01	3305



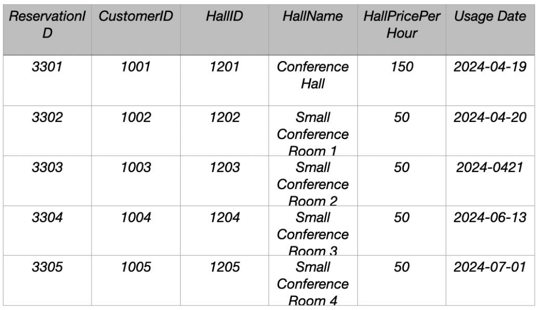
HallID	HallName	HallDescription	HallPricePerHour	ItemID	ItemName	ItemDescription	ItemPricePerHour
1201	Conference Hall	Spacious conference hall with large room	150	2001	LCD P1	Projecter	30
1202	Small Conference Room 1	Cozy small room	50	2002	LCD P2	Projecter	30
1203	Small Conference Room 2	Cozy small room	50	2003	LCD P3]	Projecter	30
1204	Small Conference Room 3	Versatile room	50	2004	PC 1	Home computer	25
1205	Small Conference Room 4	Versatile room	50	2005	PC 2	Home computer	25

3NF (Third Normal Form) is a database normalization form that builds upon 2NF and ensures that there are no transitive dependencies between non-key attributes. Transitive dependencies occur when a non-key attribute depends on another non-key attribute, which itself depends on the primary key.

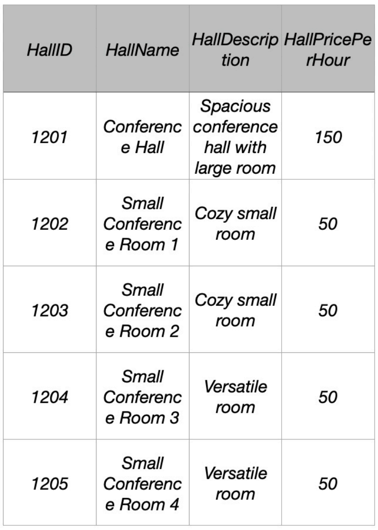


3NF

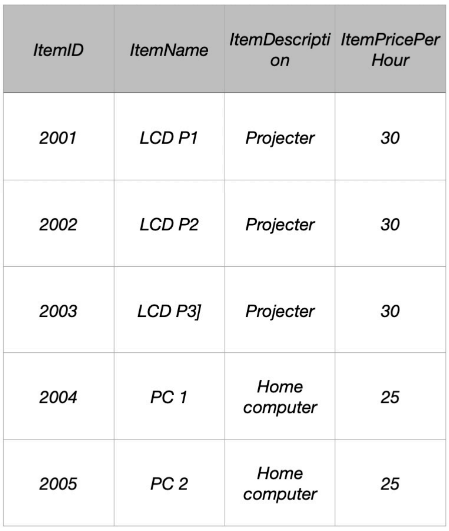
CustomerID	CustomerName	CustomerGender
1001	John Chan	M
1002	Jane Park	F
1003	Mike John	M
1004	Emma WhatsUp	F
1005	David Potter	M



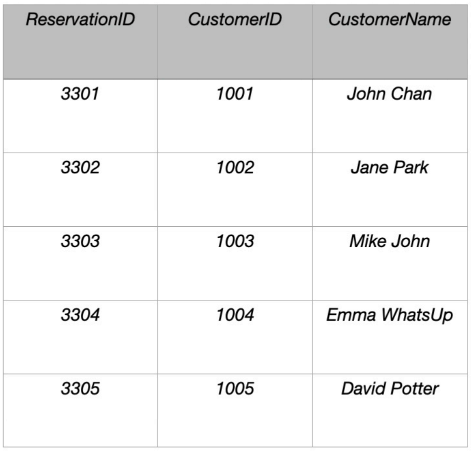
ReservationID	CustomerID	HallID	HallName	HallPricePerHour	Usage Date
3301	1001	1201	Conference Hall	150	2024-04-19
3302	1002	1202	Small Conference Room 1	50	2024-04-20
3303	1003	1203	Small Conference Room 2	50	2024-0421
3304	1004	1204	Small Conference Room 3	50	2024-06-13
3305	1005	1205	Small Conference Room 4	50	2024-07-01



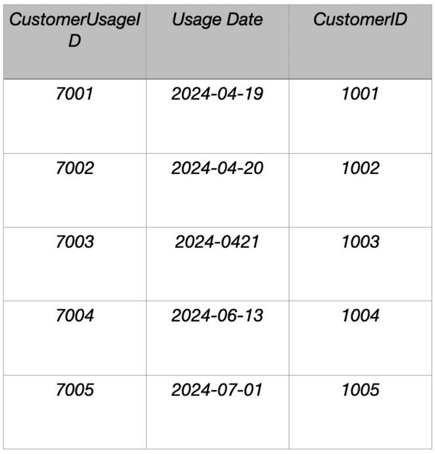
HallID	HallName	HallDescription	HallPricePerHour
1201	Conference Hall	Spacious conference hall with large room	150
1202	Small Conference Room 1	Cozy small room	50
1203	Small Conference Room 2	Cozy small room	50
1204	Small Conference Room 3	Versatile room	50
1205	Small Conference Room 4	Versatile room	50



ItemID	ItemName	ItemDescription	ItemPricePerHour
2001	LCD P1	Projecter	30
2002	LCD P2	Projecter	30
2003	LCD P3]	Projecter	30
2004	PC 1	Home computer	25
2005	PC 2	Home computer	25



ReservationID	CustomerID	CustomerName
3301	1001	John Chan
3302	1002	Jane Park
3303	1003	Mike John
3304	1004	Emma WhatsUp
3305	1005	David Potter



CustomerUsageID	Usage Date	CustomerID
7001	2024-04-19	1001
7002	2024-04-20	1002
7003	2024-0421	1003
7004	2024-06-13	1004
7005	2024-07-01	1005

7. DATA DEFINITION LANGUAGE:

TABLES:

```
-- Customer table
CREATE TABLE Customer (
CustomerID INT PRIMARY KEY,
CustomerName VARCHAR(50),
CustomerGender VARCHAR(1),
ContactNumber VARCHAR(20),
Address VARCHAR(100),
SignInDate DATE
);

-- ResourceA table
CREATE TABLE ResourceA (
HallID INT PRIMARY KEY,
HallName VARCHAR(255),
HallCapacity INT,
Location VARCHAR(255),
HallDescription VARCHAR(100),
HallPricePerHour DECIMAL(10, 2)
);
```

```
-- ResourceB table

CREATE TABLE ResourceB (

ItemID INT PRIMARY KEY,

ItemName VARCHAR(255),

ItemCapacity INT,

ItemDescription TEXT,

ItemPricePerHour DECIMAL(10, 2)

);



-- Reservation table

CREATE TABLE Reservation (

ReservationID INT PRIMARY KEY,

HallID INT,

CustomerID INT,

ReservationDate DATE,

StartTime TIME,

EndTime TIME,

FOREIGN KEY (HallID) REFERENCES ResourceA(HallID),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);
```

```
-- ResourceUtilization table
CREATE TABLE ResourceUtilization (
UtilizationID INT PRIMARY KEY,
HallID INT,
UtilizationDate DATE,
StartTime TIME,
EndTime TIME,
FOREIGN KEY (HallID) REFERENCES ResourceA(HallID)
);


-- Availability table
CREATE TABLE Availability (
AvailabilityID INT PRIMARY KEY,
HallID INT,
AvailabilityDate DATE,
StartTime TIME,
EndTime TIME,
IsAvailable BOOLEAN,
FOREIGN KEY (HallID) REFERENCES ResourceA(HallID) ON DELETE CASCADE
);
```

```
-- UsageReport table
CREATE TABLE UsageReport (
ReportID INT PRIMARY KEY,
HallID INT,
ReportDate DATE,
UtilizationDuration DECIMAL(10, 2),
FOREIGN KEY (HallID) REFERENCES ResourceA(HallID) ON DELETE CASCADE
);


-- CustomerUsage table
CREATE TABLE CustomerUsage (
UsageID INT PRIMARY KEY,
CustomerID INT,
ResourceID INT,
UsageDate DATE,
StartTime TIME,
EndTime TIME,
FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);
```

```
-- Customer table info
INSERT INTO Customer (CustomerID, CustomerName, CustomerGender, ContactNumber,
Address, SignInDate)
VALUES
(1001, 'John Chan', 'M', '11111111', '123 Tun Mun', '2022-01-22'),
(1002, 'Jane Park', 'F', '11111102', '456 HK Stree', '2022-03-01'),
(1003, 'Mike John', 'M', '11113303', '789 Oak Tree', '2022-03-02'),
(1004, 'Emma WhatsUp', 'F', '20202020', '987 Ocean Park', '2022-04-10'),
(1005, 'David Potter', 'M', '82918291', '321 Glorios Park', '2022-05-15'),
(1006, 'Sarah Brown', 'F', '67925384', '555 Universtiity', '2022-06-20'),
(1007, 'Michael Wil', 'M', '12341234', '222 Tun Mun', '2022-07-05'),
(1008, 'Jessica Me', 'F', '56785678', '999 Central Park', '2022-08-11'),
(1009, 'Daniel Kim', 'M', '13141314', '111 Kowloon', '2022-09-25'),
(1010, 'Sophia Who', 'F', '44664488', '444 Maple Street', '2022-10-30'),
(1011, 'William Dick', 'M', '89804870', '777 Tun Mun', '2022-11-14'),
(1012, 'Hugo Boss', 'F', '48848811', '999 Maple Street', '2022-12-20'),
(1013, 'O\'Mac', 'M', '34567890', '333 Tsim Sha Tsui', '2023-01-25'),
(1014, 'Emma stone', 'F', '98765432', '666 Tai Kok Tsui', '2023-02-28'),
(1015, 'Me Who', 'M', '12457800', '222 Maple Lane', '2023-03-15'),
(1016, 'Ava Eva', 'F', '55555555', '555 Shek O', '2023-04-01'),
(1017, 'Jackie Chan', 'M', '68906890', '888 Wan Chai', '2023-05-16'),
(1018, 'Mike Fu', 'F', '44449999', '689 Wan Chai', '2023-06-30'),
(1019, 'Alex Will', 'M', '56676787', '666 Mid-Levels', '2023-07-15'),
(1020, 'Sophia Chan', 'F', '88888888', '487 Tai Tam', '2023-08-31'),
(1021, 'Michael reeves', 'M', '77777777', '697 Kennedy Town', '2023-09-14'),
(1022, 'Dr Yin', 'F', '22222222', '676 Oak Tree', '2023-10-30'),
(1023, 'Supper Man', 'M', '55444787', '777 HK Street', '2023-11-12'),
```

```sql
(1024, 'Ocean Park', 'F', '55776688', '222 Quarry Bay', '2023-12-28'),
(1025, 'Tina Tifa', 'M', '89898989', '555 Ocean Park', '2024-01-02');


-- ResourceA table info
INSERT INTO ResourceA (HallID, HallName, HallCapacity, Location,
HallDescription,
HallPricePerHour)
VALUES
(1201, 'Conference Hall', 150, 'Hong Kong Central', 'Spacious conference hall
with large capacity', 150.00),
(1202, 'Small Conference Room 1', 10, 'Hong Kong Central', 'Cozy conference
room for small meetings', 50.00),
(1203, 'Small Conference Room 2', 10, 'Hong Kong Central', 'Cozy conference
room for small meetings', 50.00),
(1204, 'Small Conference Room 3', 10, 'Hong Kong Central', 'Cozy conference
room for small meetings', 50.00),
(1205, 'Small Conference Room 4', 10, 'Hong Kong Central', 'Cozy conference
room for
small meetings', 50.00),
(1206, 'Small Conference Room 5', 10, 'Hong Kong Central', 'Versatile conference
room for
medium-sized events', 50.00),
(1207, 'Small Conference Room 6', 10, 'Hong Kong Central', 'Versatile conference
room for
medium-sized events', 50.00),
(1208, 'Small Conference Room 7', 10, 'Hong Kong Central', 'Versatile conference
room for
medium-sized events', 50.00),
(1209, 'Small Conference Room 8', 10, 'Hong Kong Central', 'Versatile conference
room for
medium-sized events', 50.00),
(1210, 'Small Conference Room 9', 10, 'Hong Kong Central', 'Cozy conference room for
small meetings', 50.00),
(1211, 'Small Conference Room 10', 10, 'Hong Kong Central', 'Cozy conference room for
small meetings', 50.00),
(1212, 'Small Conference Room 11', 10, 'Hong Kong Central', 'Cozy conference room for
small meetings', 50.00);



-- ResourceB table info
INSERT INTO ResourceB (ItemID, ItemName, ItemCapacity, ItemDescription, ItemPricePerHour)
VALUES
(2001, 'LCD Projector01', 1, 'High-quality projector for presentations', 30.00),
(2002, 'LCD Prohector02', 1, 'High-quality projector for presentations', 30.00),
(2003, 'LCD Prohector03', 1, 'High-quality projector for presentations', 30.00),
(2004, 'LCD Prohector04', 1, 'High-quality projector for presentations', 30.00),
(2005, 'LCD Projector05', 1, 'High-quality projector for presentations', 30.00),
(2006, 'LCD Projector06', 1, 'High-quality projector for presentations', 30.00),
(2007, 'LCD Projector07', 1, 'High-quality projector for presentations', 30.00),
(2008, 'Portable PC01', 1, 'Lightweight and easy to carry for meeting and work', 25.00),
(2009, 'Portable PC02', 1, 'Lightweight and easy to carry for meeting and work', 25.00),
(2010, 'Portable PC03', 1, 'Lightweight and easy to carry for meeting and work', 25.00),
(2011, 'Portable PC04', 1, 'Lightweight and easy to carry for meeting and work', 25.00),

(2012, 'Portable PC05', 1, 'Lightweight and easy to carry for meeting and work', 25.00);


-- Availability table info

INSERT INTO Availability (AvailabilityID, HallID, AvailabilityDate, StartTime, EndTime, IsAvailable)

VALUES

(5001, 1201, '2024-04-19', '09:00:00', '11:00:00', TRUE),

(5002, 1201, '2024-04-20', '14:00:00', '16:00:00', TRUE),

(5003, 1206, '2024-04-21', '10:00:00', '12:00:00', TRUE),

(5004, 1207, '2024-04-22', '13:00:00', '15:00:00', TRUE),

(5005, 1207, '2024-04-23', '09:00:00', '11:00:00', TRUE),

(5006, 1212, '2024-04-24', '14:00:00', '16:00:00', TRUE),

(5007, 2001, '2024-04-25', '10:00:00', '12:00:00', TRUE),

(5008, 2008, '2024-04-26', '13:00:00', '15:00:00', TRUE);


-- UsageReport table info

INSERT INTO UsageReport (ReportID, HallID, ReportDate, UtilizationDuration)

VALUES

(6001, 2001, '2024-04-19', 2.5),

(6002, 2002, '2024-04-20', 1.75),

(6003, 2003, '2024-04-21', 2.0),

(6004, 2004, '2024-04-22', 1.5),

(6005, 2005, '2024-04-23', 2.25),

(6006, 2006, '2024-04-24', 1.75),

(6007, 2007, '2024-04-25', 2.0),

(6008, 2008, '2024-04-26', 1.5);
```

```
-- CustomerUsage table info
INSERT INTO CustomerUsage (UsageID, CustomerID, ResourceID, UsageDate,
StartTime, EndTime)
VALUES
(7001, 1021, 1201, '2024-04-19', '09:00:00', '11:00:00'),
(7002, 1001, 2002, '2024-04-20', '14:00:00', '16:00:00'),
(7003, 1001, 2002, '2024-04-21', '10:00:00', '12:00:00'),
(7004, 1013, 2004, '2024-04-22', '13:00:00', '15:00:00'),
(7005, 1004, 1201, '2024-04-23', '16:00:00', '18:00:00'),
(7006, 1007, 2006, '2024-04-24', '09:00:00', '11:00:00'),
(7007, 1017, 2007, '2024-04-25', '14:00:00', '16:00:00'),
(7008, 1022, 1211, '2024-04-26', '10:00:00', '12:00:00'),
(7009, 1011, 2009, '2024-04-27', '11:00:00', '13:00:00');
```

```
-- Reservation table info
INSERT INTO Reservation (ReservationID, HallID, CustomerID, ReservationDate,
StartTime, EndTime)
VALUES
(3301, 1201, 1001, '2024-04-19', '09:00:00', '11:00:00'),
(3302, 1201, 1001, '2024-04-20', '14:00:00', '16:00:00'),
(3303, 1206, 1003, '2024-04-21', '10:00:00', '12:00:00'),
(3404, 1204, 1004, '2024-04-28', '18:00:00', '22:00:00'),
(3505, 1205, 1005, '2024-04-29', '13:00:00', '15:00:00'),
(3606, 1203, 1006, '2024-05-01', '16:00:00', '18:00:00');
```

```
-- ResourceUtilization table info

INSERT INTO ResourceUtilization (UtilizationID, HallID, UtilizationDate, StartTime, EndTime)

VALUES

(3001, 1201, '2024-04-19', '09:00:00', '11:00:00'),

(3002, 1202, '2024-04-20', '14:00:00', '16:00:00'),

(3003, 1206, '2024-04-21', '10:00:00', '12:00:00'),

(3004, 1204, '2024-04-28', '18:00:00', '22:00:00'),

(3005, 1205, '2024-04-29', '13:00:00', '15:00:00'),

(3006, 1203, '2024-05-01', '16:00:00', '18:00:00');
```

SCREENSHOTS OF TABLES:

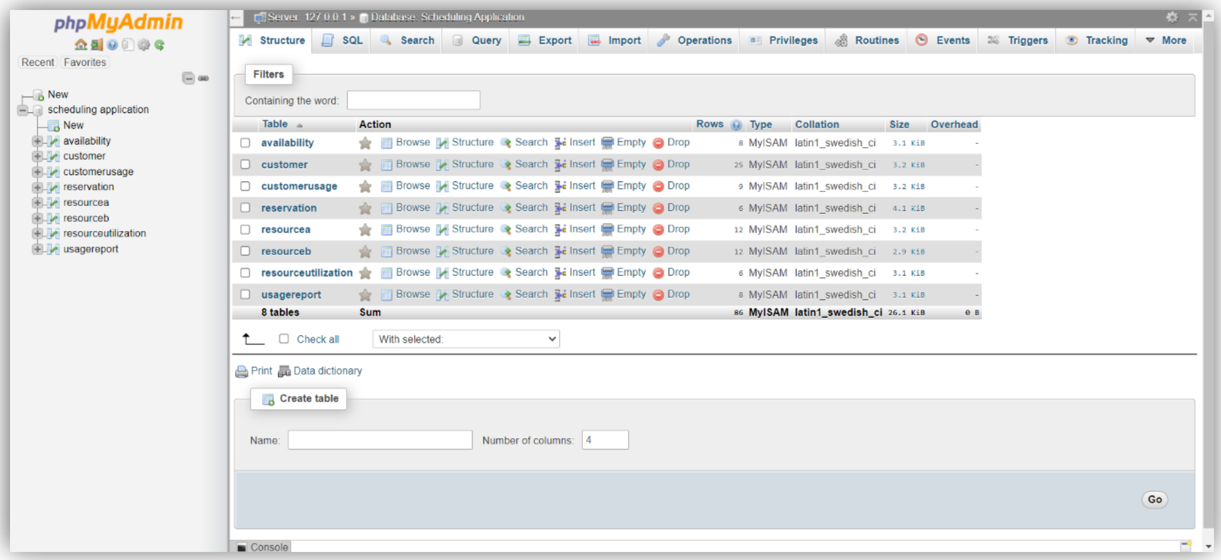
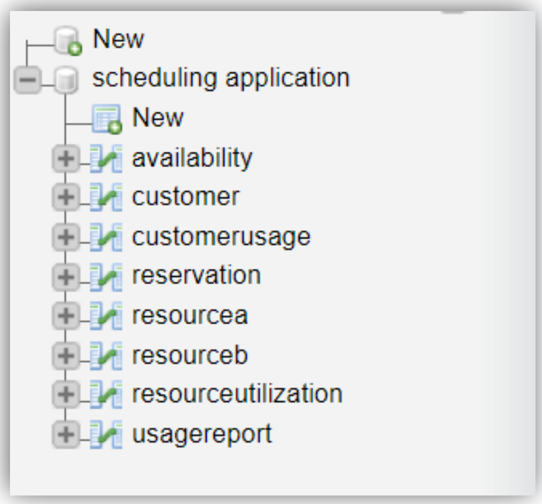
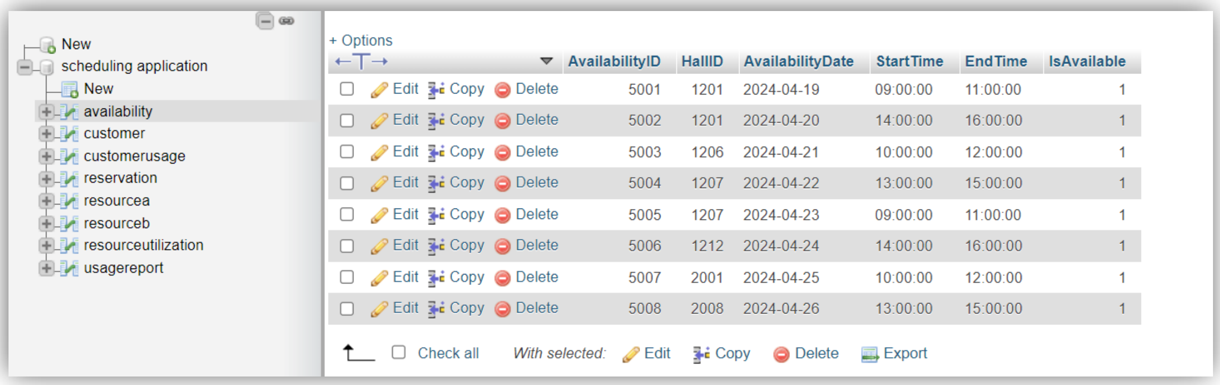


	Table	Action	Rows	Type	Collation	Size	Overhead
□	availability	Browse Structure Search Insert Empty Drop	8	MyISAM	latin1_swedish_ci	3.1 KiB	-
□	customer	Browse Structure Search Insert Empty Drop	25	MyISAM	latin1_swedish_ci	3.2 KiB	-
□	customerusage	Browse Structure Search Insert Empty Drop	9	MyISAM	latin1_swedish_ci	3.2 KiB	-
□	reservation	Browse Structure Search Insert Empty Drop	6	MyISAM	latin1_swedish_ci	4.1 KiB	-
□	resourcea	Browse Structure Search Insert Empty Drop	12	MyISAM	latin1_swedish_ci	3.2 KiB	-
□	resourceb	Browse Structure Search Insert Empty Drop	12	MyISAM	latin1_swedish_ci	2.9 KiB	-
□	resourceutilization	Browse Structure Search Insert Empty Drop	6	MyISAM	latin1_swedish_ci	3.1 KiB	-
□	usagereport	Browse Structure Search Insert Empty Drop	8	MyISAM	latin1_swedish_ci	3.1 KiB	-
	8 tables	Sum	86	MyISAM	latin1_swedish_ci	26.1 KiB	0 B

□ Check all With selected:

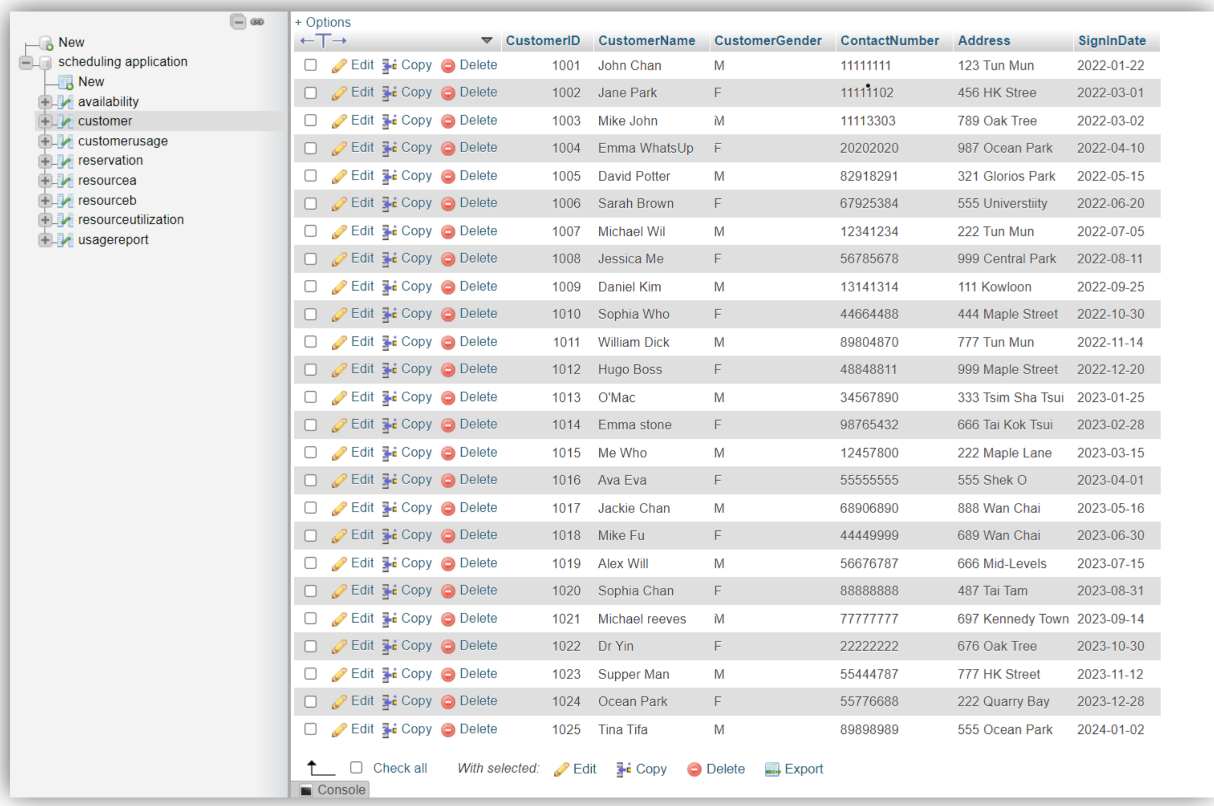


New
scheduling application
New
availability
customer
customerusage
reservation
resourcea
resourceb
resourceutilization
usagereport

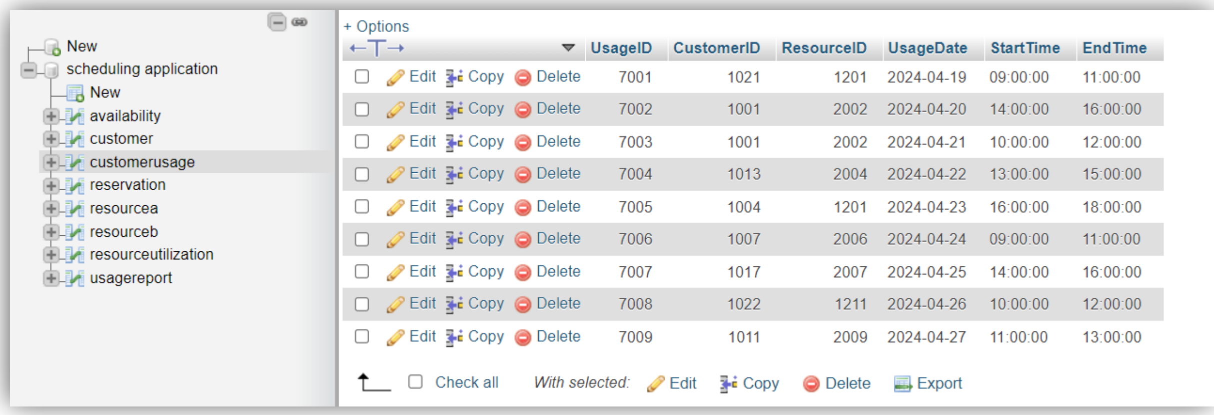


		AvailabilityID	HallID	AvailabilityDate	StartTime	EndTime	IsAvailable
□	Edit Copy Delete	5001	1201	2024-04-19	09:00:00	11:00:00	1
□	Edit Copy Delete	5002	1201	2024-04-20	14:00:00	16:00:00	1
□	Edit Copy Delete	5003	1206	2024-04-21	10:00:00	12:00:00	1
□	Edit Copy Delete	5004	1207	2024-04-22	13:00:00	15:00:00	1
□	Edit Copy Delete	5005	1207	2024-04-23	09:00:00	11:00:00	1
□	Edit Copy Delete	5006	1212	2024-04-24	14:00:00	16:00:00	1
□	Edit Copy Delete	5007	2001	2024-04-25	10:00:00	12:00:00	1
□	Edit Copy Delete	5008	2008	2024-04-26	13:00:00	15:00:00	1

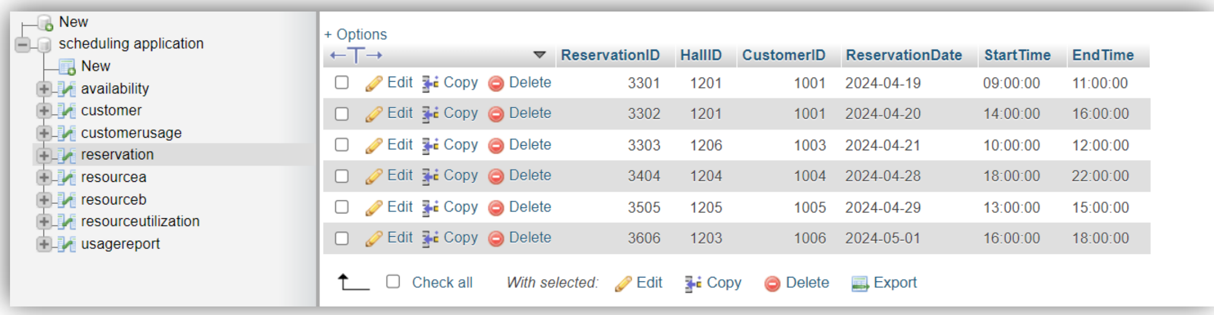
□ Check all *With selected:* Edit Copy Delete Export



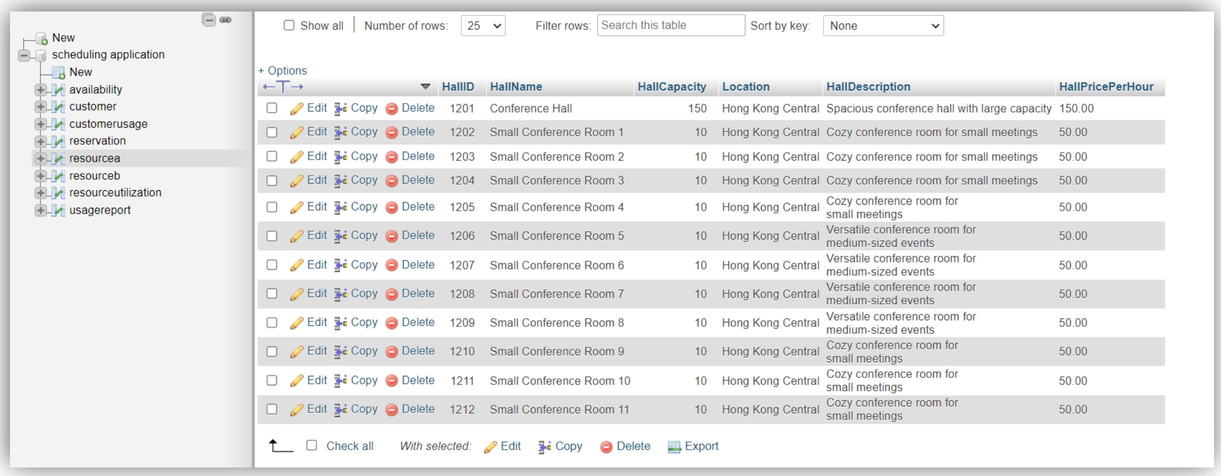
	CustomerID	CustomerName	CustomerGender	ContactNumber	Address	SignInDate
□ Edit Copy Delete	1001	John Chan	M	11111111	123 Tun Mun	2022-01-22
□ Edit Copy Delete	1002	Jane Park	F	11111102	456 HK Stree	2022-03-01
□ Edit Copy Delete	1003	Mike John	M	11113303	789 Oak Tree	2022-03-02
□ Edit Copy Delete	1004	Emma WhatsUp	F	20202020	987 Ocean Park	2022-04-10
□ Edit Copy Delete	1005	David Potter	M	82918291	321 Glorios Park	2022-05-15
□ Edit Copy Delete	1006	Sarah Brown	F	67925384	555 Universtiity	2022-06-20
□ Edit Copy Delete	1007	Michael Wil	M	12341234	222 Tun Mun	2022-07-05
□ Edit Copy Delete	1008	Jessica Me	F	56785678	999 Central Park	2022-08-11
□ Edit Copy Delete	1009	Daniel Kim	M	13141314	111 Kowloon	2022-09-25
□ Edit Copy Delete	1010	Sophia Who	F	44664488	444 Maple Street	2022-10-30
□ Edit Copy Delete	1011	William Dick	M	89804870	777 Tun Mun	2022-11-14
□ Edit Copy Delete	1012	Hugo Boss	F	48848811	999 Maple Street	2022-12-20
□ Edit Copy Delete	1013	O'Mac	M	34567890	333 Tsim Sha Tsui	2023-01-25
□ Edit Copy Delete	1014	Emma stone	F	98765432	666 Tai Kok Tsui	2023-02-28
□ Edit Copy Delete	1015	Me Who	M	12457800	222 Maple Lane	2023-03-15
□ Edit Copy Delete	1016	Ava Eva	F	55555555	555 Shek O	2023-04-01
□ Edit Copy Delete	1017	Jackie Chan	M	68906890	888 Wan Chai	2023-05-16
□ Edit Copy Delete	1018	Mike Fu	F	44449999	689 Wan Chai	2023-06-30
□ Edit Copy Delete	1019	Alex Will	M	56676787	666 Mid-Levels	2023-07-15
□ Edit Copy Delete	1020	Sophia Chan	F	88888888	487 Tai Tam	2023-08-31
□ Edit Copy Delete	1021	Michael reeves	M	77777777	697 Kennedy Town	2023-09-14
□ Edit Copy Delete	1022	Dr Yin	F	22222222	676 Oak Tree	2023-10-30
□ Edit Copy Delete	1023	Supper Man	M	55444787	777 HK Street	2023-11-12
□ Edit Copy Delete	1024	Ocean Park	F	55776688	222 Quarry Bay	2023-12-28
□ Edit Copy Delete	1025	Tina Tifa	M	89898989	555 Ocean Park	2024-01-02



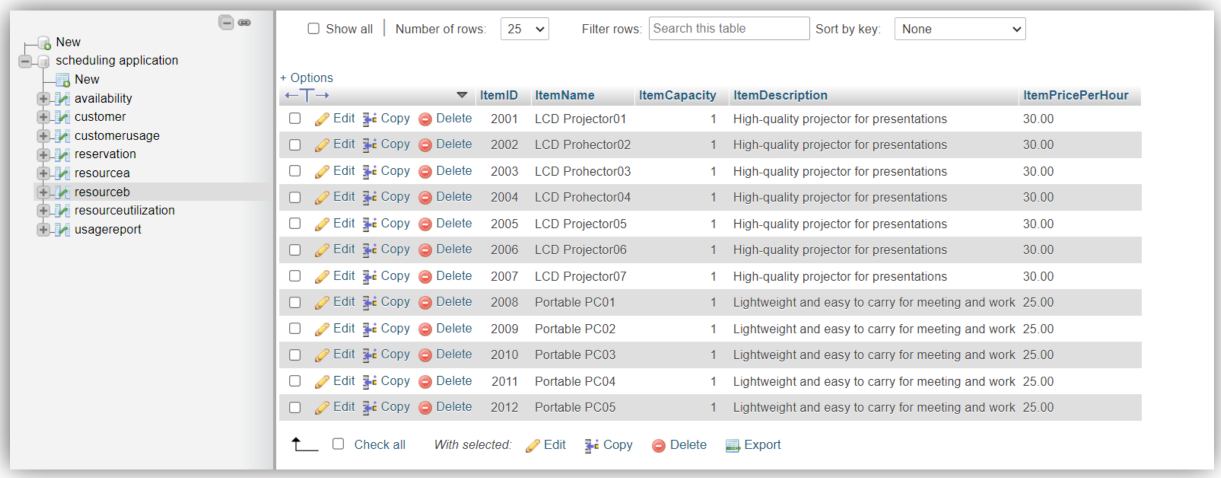
	UsageID	CustomerID	ResourceID	UsageDate	StartTime	EndTime
□ Edit Copy Delete	7001	1021	1201	2024-04-19	09:00:00	11:00:00
□ Edit Copy Delete	7002	1001	2002	2024-04-20	14:00:00	16:00:00
□ Edit Copy Delete	7003	1001	2002	2024-04-21	10:00:00	12:00:00
□ Edit Copy Delete	7004	1013	2004	2024-04-22	13:00:00	15:00:00
□ Edit Copy Delete	7005	1004	1201	2024-04-23	16:00:00	18:00:00
□ Edit Copy Delete	7006	1007	2006	2024-04-24	09:00:00	11:00:00
□ Edit Copy Delete	7007	1017	2007	2024-04-25	14:00:00	16:00:00
□ Edit Copy Delete	7008	1022	1211	2024-04-26	10:00:00	12:00:00
□ Edit Copy Delete	7009	1011	2009	2024-04-27	11:00:00	13:00:00



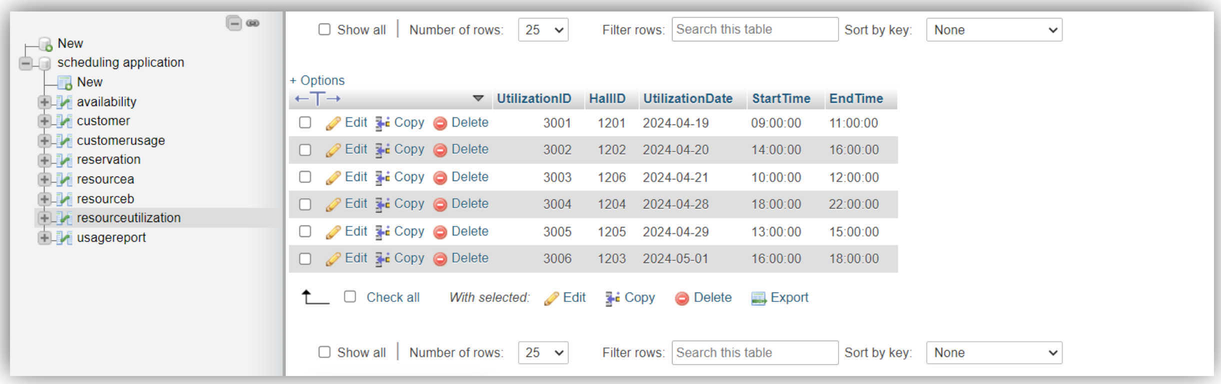
	ReservationID	HallID	CustomerID	ReservationDate	StartTime	EndTime
□ Edit Copy Delete	3301	1201	1001	2024-04-19	09:00:00	11:00:00
□ Edit Copy Delete	3302	1201	1001	2024-04-20	14:00:00	16:00:00
□ Edit Copy Delete	3303	1206	1003	2024-04-21	10:00:00	12:00:00
□ Edit Copy Delete	3404	1204	1004	2024-04-28	18:00:00	22:00:00
□ Edit Copy Delete	3505	1205	1005	2024-04-29	13:00:00	15:00:00
□ Edit Copy Delete	3606	1203	1006	2024-05-01	16:00:00	18:00:00



	HallID	HallName	HallCapacity	Location	HallDescription	HallPricePerHour
□ Edit Copy Delete	1201	Conference Hall	150	Hong Kong Central	Spacious conference hall with large capacity	150.00
□ Edit Copy Delete	1202	Small Conference Room 1	10	Hong Kong Central	Cozy conference room for small meetings	50.00
□ Edit Copy Delete	1203	Small Conference Room 2	10	Hong Kong Central	Cozy conference room for small meetings	50.00
□ Edit Copy Delete	1204	Small Conference Room 3	10	Hong Kong Central	Cozy conference room for small meetings	50.00
□ Edit Copy Delete	1205	Small Conference Room 4	10	Hong Kong Central	Cozy conference room for small meetings	50.00
□ Edit Copy Delete	1206	Small Conference Room 5	10	Hong Kong Central	Versatile conference room for medium-sized events	50.00
□ Edit Copy Delete	1207	Small Conference Room 6	10	Hong Kong Central	Versatile conference room for medium-sized events	50.00
□ Edit Copy Delete	1208	Small Conference Room 7	10	Hong Kong Central	Versatile conference room for medium-sized events	50.00
□ Edit Copy Delete	1209	Small Conference Room 8	10	Hong Kong Central	Versatile conference room for medium-sized events	50.00
□ Edit Copy Delete	1210	Small Conference Room 9	10	Hong Kong Central	Cozy conference room for small meetings	50.00
□ Edit Copy Delete	1211	Small Conference Room 10	10	Hong Kong Central	Cozy conference room for small meetings	50.00
□ Edit Copy Delete	1212	Small Conference Room 11	10	Hong Kong Central	Cozy conference room for small meetings	50.00

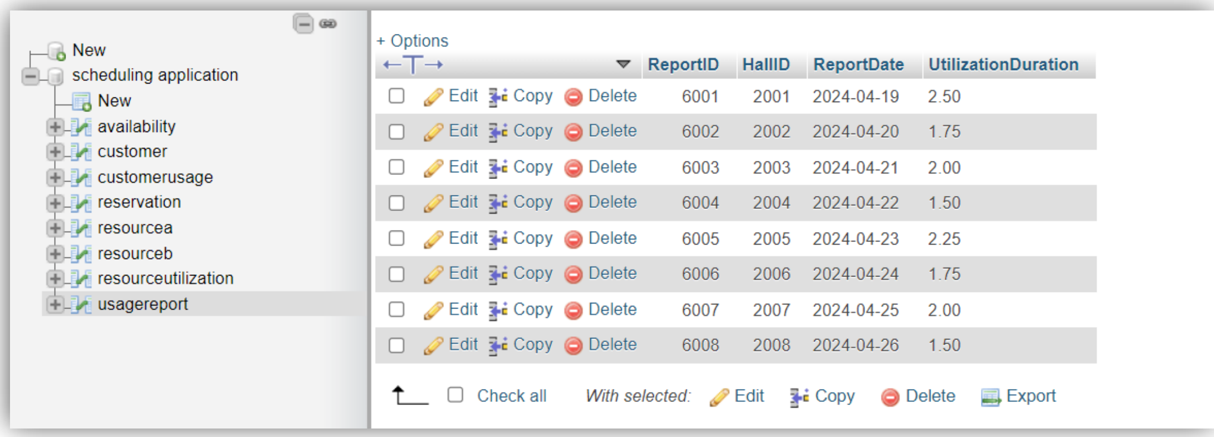


	ItemID	ItemName	ItemCapacity	ItemDescription	ItemPricePerHour
□ Edit Copy Delete	2001	LCD Projector01	1	High-quality projector for presentations	30.00
□ Edit Copy Delete	2002	LCD Prohector02	1	High-quality projector for presentations	30.00
□ Edit Copy Delete	2003	LCD Prohector03	1	High-quality projector for presentations	30.00
□ Edit Copy Delete	2004	LCD Prohector04	1	High-quality projector for presentations	30.00
□ Edit Copy Delete	2005	LCD Projector05	1	High-quality projector for presentations	30.00
□ Edit Copy Delete	2006	LCD Projector06	1	High-quality projector for presentations	30.00
□ Edit Copy Delete	2007	LCD Projector07	1	High-quality projector for presentations	30.00
□ Edit Copy Delete	2008	Portable PC01	1	Lightweight and easy to carry for meeting and work	25.00
□ Edit Copy Delete	2009	Portable PC02	1	Lightweight and easy to carry for meeting and work	25.00
□ Edit Copy Delete	2010	Portable PC03	1	Lightweight and easy to carry for meeting and work	25.00
□ Edit Copy Delete	2011	Portable PC04	1	Lightweight and easy to carry for meeting and work	25.00
□ Edit Copy Delete	2012	Portable PC05	1	Lightweight and easy to carry for meeting and work	25.00



	UtilizationID	HallID	UtilizationDate	StartTime	EndTime
□ Edit Copy Delete	3001	1201	2024-04-19	09:00:00	11:00:00
□ Edit Copy Delete	3002	1202	2024-04-20	14:00:00	16:00:00
□ Edit Copy Delete	3003	1206	2024-04-21	10:00:00	12:00:00
□ Edit Copy Delete	3004	1204	2024-04-28	18:00:00	22:00:00
□ Edit Copy Delete	3005	1205	2024-04-29	13:00:00	15:00:00
□ Edit Copy Delete	3006	1203	2024-05-01	16:00:00	18:00:00

□ Check all
With selected: Edit Copy Delete Export
□ Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None

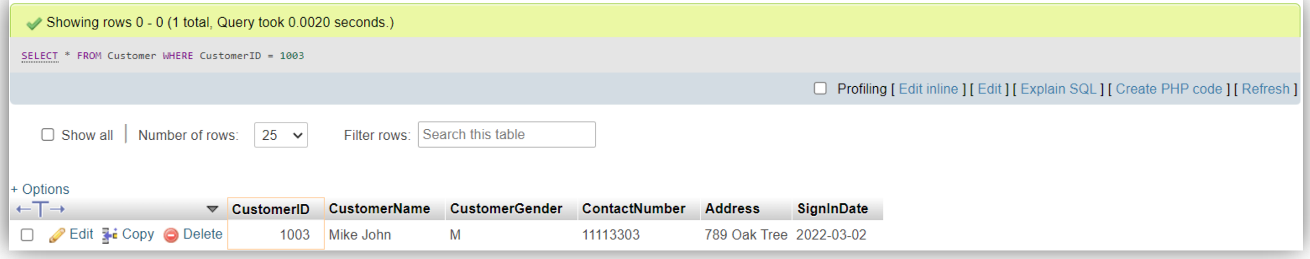


	ReportID	HallID	ReportDate	UtilizationDuration
□ Edit Copy Delete	6001	2001	2024-04-19	2.50
□ Edit Copy Delete	6002	2002	2024-04-20	1.75
□ Edit Copy Delete	6003	2003	2024-04-21	2.00
□ Edit Copy Delete	6004	2004	2024-04-22	1.50
□ Edit Copy Delete	6005	2005	2024-04-23	2.25
□ Edit Copy Delete	6006	2006	2024-04-24	1.75
□ Edit Copy Delete	6007	2007	2024-04-25	2.00
□ Edit Copy Delete	6008	2008	2024-04-26	1.50

8. SQL QUERIES:

❖ Retrieve the customer details for the customer with CustomerID 1003.

```
SELECT *
FROM Customer
WHERE CustomerID = 1003;
```


Showing rows 0 - 0 (1 total, Query took 0.0020 seconds.)

```
SELECT * FROM Customer WHERE CustomerID = 1003
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
+ Options

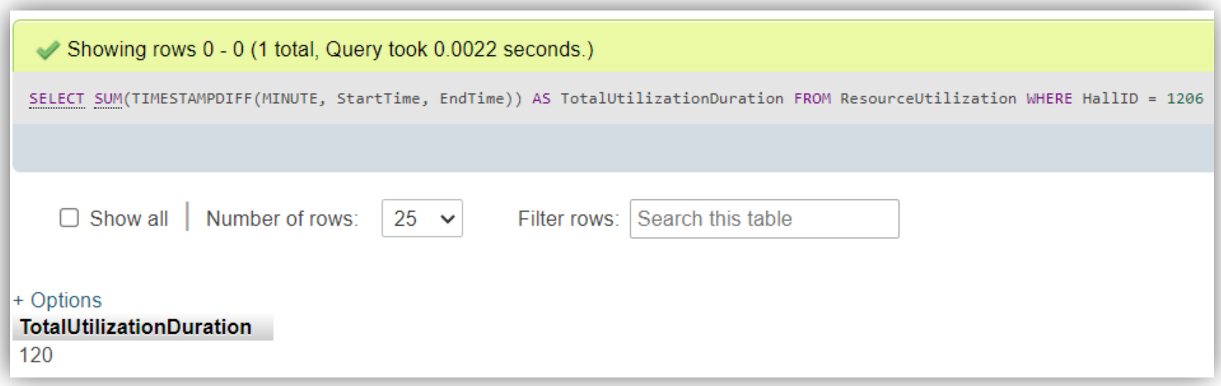
	CustomerID	CustomerName	CustomerGender	ContactNumber	Address	SignInDate
□ Edit Copy Delete	1003	Mike John	M	11113303	789 Oak Tree	2022-03-02

❖ What is the total utilization duration for ResourceA hall with HallID 1206?

SELECT SUM(TIMESTAMPDIFF(MINUTE, StartTime, EndTime)) AS TotalUtilizationDuration

FROM ResourceUtilization

WHERE HallID = 1206;



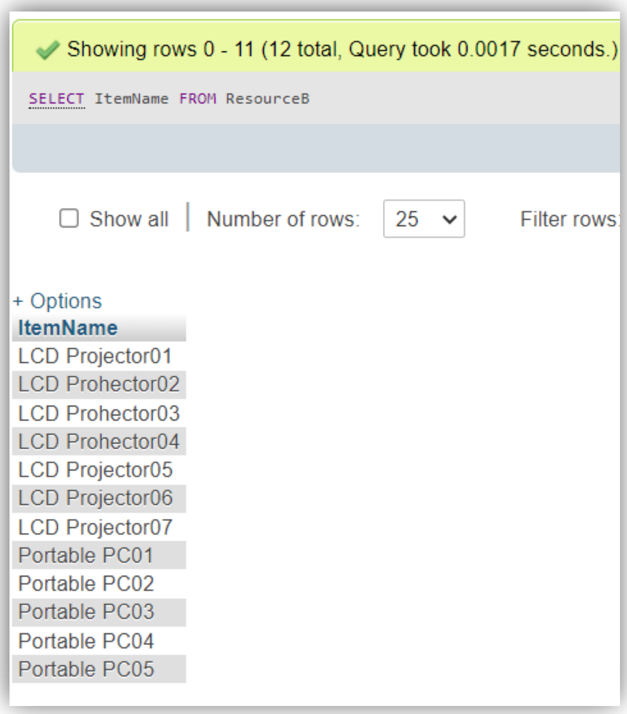
✓ Showing rows 0 - 0 (1 total, Query took 0.0022 seconds.)

```
SELECT SUM(TIMESTAMPDIFF(MINUTE, StartTime, EndTime)) AS TotalUtilizationDuration FROM ResourceUtilization WHERE HallID = 1206
```

TotalUtilizationDuration
120

❖ List all the resources (itemName) available in ResourceB table.

SELECT ItemName
FROM ResourceB;



✓ Showing rows 0 - 11 (12 total, Query took 0.0017 seconds.)

```
SELECT ItemName FROM ResourceB
```

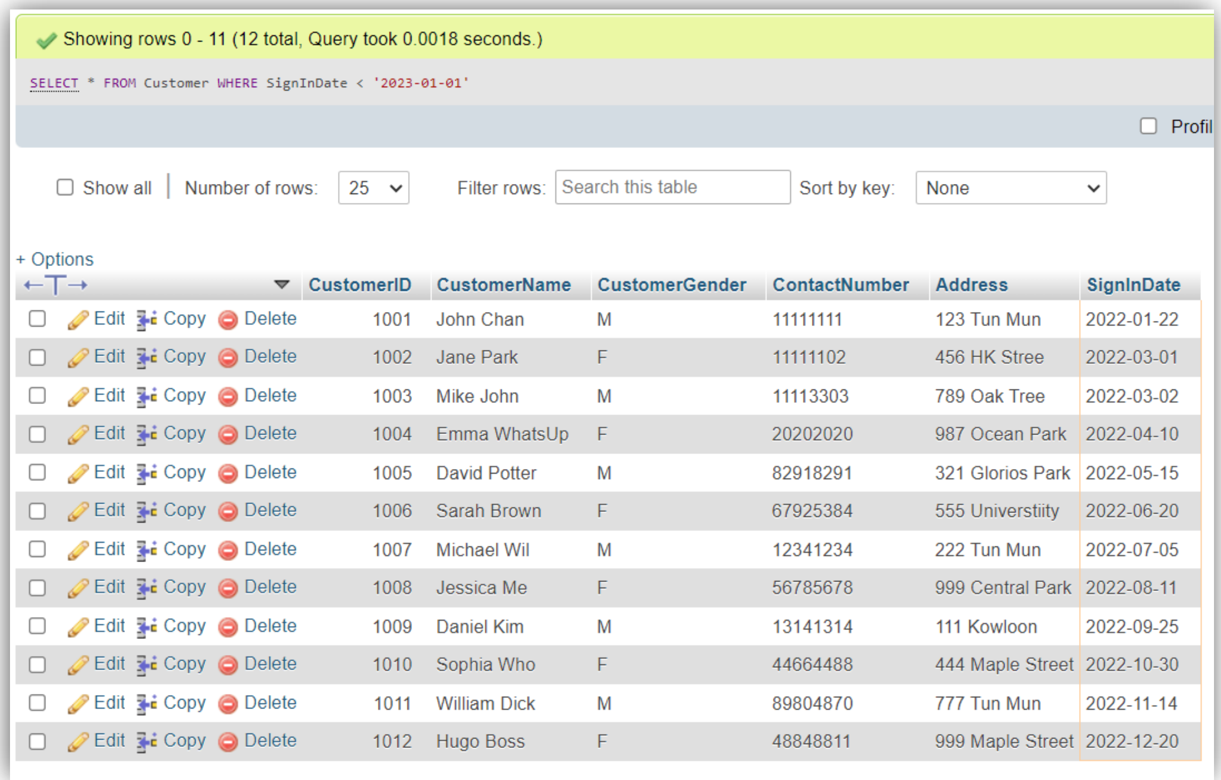
ItemName
LCD Projector01
LCD Prohector02
LCD Prohector03
LCD Prohector04
LCD Projector05
LCD Projector06
LCD Projector07
Portable PC01
Portable PC02
Portable PC03
Portable PC04
Portable PC05

- Retrieve the customer details for customers who signed in before ‘2023-01-01’.

```
SELECT *

FROM Customer

WHERE SignInDate < '2023-01-01';
```



✓ Showing rows 0 - 11 (12 total, Query took 0.0018 seconds.)

```
SELECT * FROM Customer WHERE SignInDate < '2023-01-01'
```

□ Profil

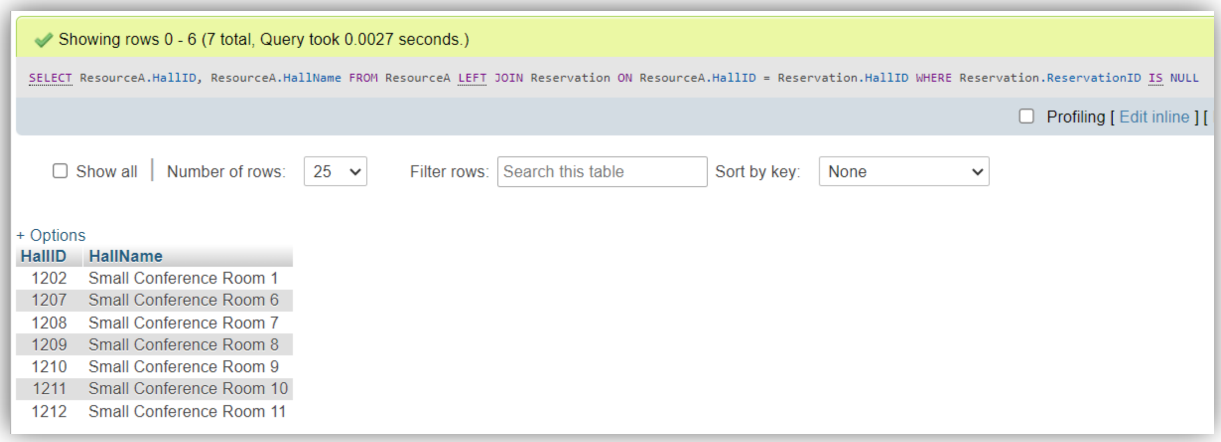
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	CustomerID	CustomerName	CustomerGender	ContactNumber	Address	SignInDate
□ Edit Copy Delete	1001	John Chan	M	11111111	123 Tun Mun	2022-01-22
□ Edit Copy Delete	1002	Jane Park	F	11111102	456 HK Stree	2022-03-01
□ Edit Copy Delete	1003	Mike John	M	11113303	789 Oak Tree	2022-03-02
□ Edit Copy Delete	1004	Emma WhatsUp	F	20202020	987 Ocean Park	2022-04-10
□ Edit Copy Delete	1005	David Potter	M	82918291	321 Glorios Park	2022-05-15
□ Edit Copy Delete	1006	Sarah Brown	F	67925384	555 Universtiity	2022-06-20
□ Edit Copy Delete	1007	Michael Wil	M	12341234	222 Tun Mun	2022-07-05
□ Edit Copy Delete	1008	Jessica Me	F	56785678	999 Central Park	2022-08-11
□ Edit Copy Delete	1009	Daniel Kim	M	13141314	111 Kowloon	2022-09-25
□ Edit Copy Delete	1010	Sophia Who	F	44664488	444 Maple Street	2022-10-30
□ Edit Copy Delete	1011	William Dick	M	89804870	777 Tun Mun	2022-11-14
□ Edit Copy Delete	1012	Hugo Boss	F	48848811	999 Maple Street	2022-12-20

- List the resource halls that have never been reserved.

```
SELECT ResourceA.HallID, ResourceA.HallName
FROM ResourceA
LEFT JOIN Reservation ON ResourceA.HallID = Reservation.HallID
WHERE Reservation.ReservationID IS NULL;
```



✓ Showing rows 0 - 6 (7 total, Query took 0.0027 seconds.)

```
SELECT ResourceA.HallID, ResourceA.HallName FROM ResourceA LEFT JOIN Reservation ON ResourceA.HallID = Reservation.HallID WHERE Reservation.ReservationID IS NULL
```

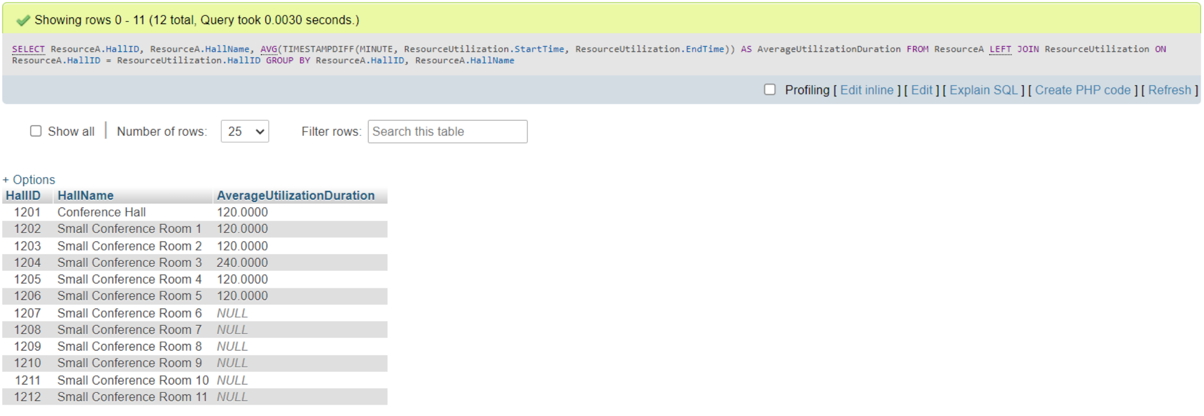
□ Profiling [Edit inline] [

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

HallID	HallName
1202	Small Conference Room 1
1207	Small Conference Room 6
1208	Small Conference Room 7
1209	Small Conference Room 8
1210	Small Conference Room 9
1211	Small Conference Room 10
1212	Small Conference Room 11

- Find the average utilization duration for each resource hall.

```
SELECT ResourceA.HallID, ResourceA.HallName, AVG(TIMESTAMPDIFF(MINUTE,
ResourceUtilization.StartTime, ResourceUtilization.EndTime)) AS
AverageUtilizationDuration
FROM ResourceA
LEFT JOIN ResourceUtilization ON ResourceA.HallID = ResourceUtilization.HallID
GROUP BY ResourceA.HallID, ResourceA.HallName;
```



✓ Showing rows 0 - 11 (12 total, Query took 0.0030 seconds.)

```
SELECT ResourceA.HallID, ResourceA.HallName, AVG(TIMESTAMPDIFF(MINUTE, ResourceUtilization.StartTime, ResourceUtilization.EndTime)) AS AverageUtilizationDuration FROM ResourceA LEFT JOIN ResourceUtilization ON
ResourceA.HallID = ResourceUtilization.HallID GROUP BY ResourceA.HallID, ResourceA.HallName
```

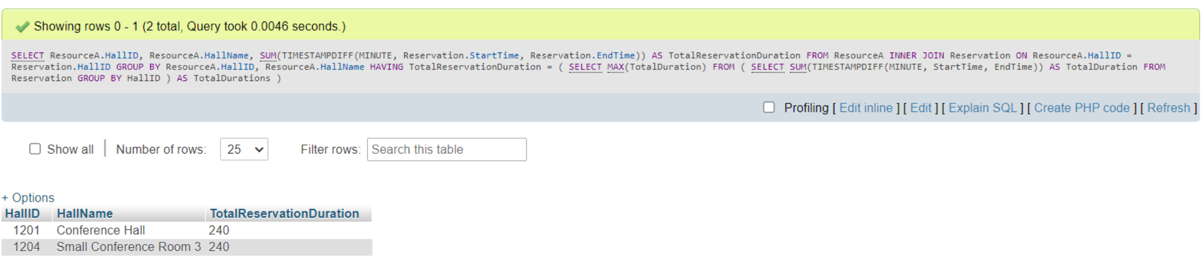
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

HallID	HallName	AverageUtilizationDuration
1201	Conference Hall	120.0000
1202	Small Conference Room 1	120.0000
1203	Small Conference Room 2	120.0000
1204	Small Conference Room 3	240.0000
1205	Small Conference Room 4	120.0000
1206	Small Conference Room 5	120.0000
1207	Small Conference Room 6	*NULL*
1208	Small Conference Room 7	*NULL*
1209	Small Conference Room 8	*NULL*
1210	Small Conference Room 9	*NULL*
1211	Small Conference Room 10	*NULL*
1212	Small Conference Room 11	*NULL*

- Retrieve the resource hall(s) that have the highest total reservation duration.

```
SELECT ResourceA.HallID, ResourceA.HallName,
SUM(TIMESTAMPDIFF(MINUTE, Reservation.StartTime, Reservation.EndTime)) AS
TotalReservationDuration
FROM ResourceA
INNER JOIN Reservation ON ResourceA.HallID = Reservation.HallID
GROUP BY ResourceA.HallID, ResourceA.HallName
HAVING TotalReservationDuration = (
    SELECT MAX(TotalDuration)
    FROM (
        SELECT SUM(TIMESTAMPDIFF(MINUTE, StartTime, EndTime)) AS
TotalDuration
        FROM Reservation
        GROUP BY HallID
    ) AS TotalDurations
);
```



✓ Showing rows 0 - 1 (2 total, Query took 0.0046 seconds.)

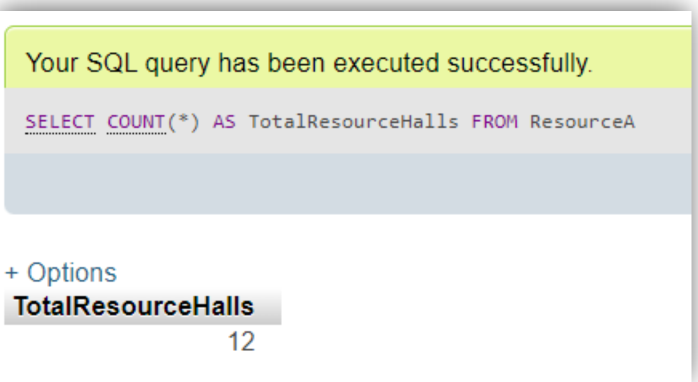
```
SELECT ResourceA.HallID, ResourceA.HallName, SUM(TIMESTAMPDIFF(MINUTE, Reservation.StartTime, Reservation.EndTime)) AS TotalReservationDuration FROM ResourceA INNER JOIN Reservation ON ResourceA.HallID =
Reservation.HallID GROUP BY ResourceA.HallID, ResourceA.HallName HAVING TotalReservationDuration = ( SELECT MAX(TotalDuration) FROM ( SELECT SUM(TIMESTAMPDIFF(MINUTE, StartTime, EndTime)) AS TotalDuration FROM
Reservation GROUP BY HallID ) AS TotalDurations )
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
+ Options

HallID	HallName	TotalReservationDuration
1201	Conference Hall	240
1204	Small Conference Room 3	240

- How many resource halls are available in the database?

```
SELECT COUNT(*) AS TotalResourceHalls
FROM ResourceA;
```



Your SQL query has been executed successfully.

```
SELECT COUNT(*) AS TotalResourceHalls FROM ResourceA
```

TotalResourceHalls
12

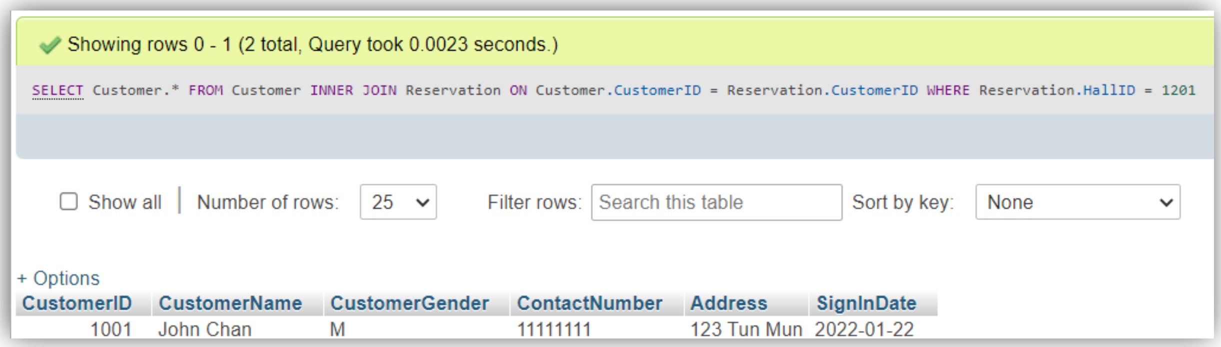
❖ Which customers have made reservations for ResourceA hall with HallID 1201?

```
SELECT Customer.*

FROM Customer

INNER JOIN Reservation ON Customer.CustomerID = Reservation.CustomerID

WHERE Reservation.HallID = 1201;
```



✓ Showing rows 0 - 1 (2 total, Query took 0.0023 seconds.)

```
SELECT Customer.* FROM Customer INNER JOIN Reservation ON Customer.CustomerID = Reservation.CustomerID WHERE Reservation.HallID = 1201
```

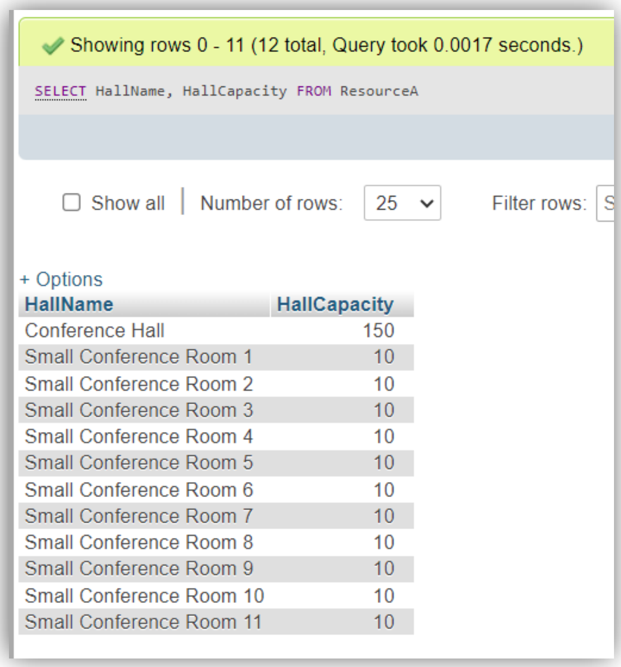
☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

CustomerID	CustomerName	CustomerGender	ContactNumber	Address	SignInDate
1001	John Chan	M	11111111	123 Tun Mun	2022-01-22

❖ List the names and capacities of all resource halls.

```
SELECT HallName, HallCapacity
FROM ResourceA;
```



✓ Showing rows 0 - 11 (12 total, Query took 0.0017 seconds.)

```
SELECT HallName, HallCapacity FROM ResourceA
```

☐ Show all | Number of rows: 25 Filter rows: S

+ Options

HallName	HallCapacity
Conference Hall	150
Small Conference Room 1	10
Small Conference Room 2	10
Small Conference Room 3	10
Small Conference Room 4	10
Small Conference Room 5	10
Small Conference Room 6	10
Small Conference Room 7	10
Small Conference Room 8	10
Small Conference Room 9	10
Small Conference Room 10	10
Small Conference Room 11	10

9. CONCLUSION:

a) Throughout the development of the Scheduling Application for the coworking space company, our group encountered various challenges and gained valuable insights. One of the most difficult aspects was ensuring the avoidance of conflicting resource usage while scheduling. Balancing the availability and demand for different resources proved to be a complex task requiring careful consideration of time slots and resource dependencies. Additionally, implementing the reporting functionality to track resource utilization per week, month, and year presented its own set of challenges, as it involved aggregating and analyzing large amounts of data.

Throughout the project, we learned the significance of proper data modeling and database design. It became evident that a well-structured database schema and efficient query design were crucial for optimizing performance and ensuring accurate reporting. We also gained a deeper understanding of resource management and the complexities involved in maintaining an efficient scheduling system.

If we were to approach the project again, we would focus on enhancing the system's scalability and flexibility. Considering the dynamic nature of resource availability and customer demand, incorporating real-time updates and advanced algorithms for resource allocation could further improve the application.

b) The proposed benefits of the new Scheduling Application are indeed achievable. By centralizing and automating the process of scheduling resources, the coworking space company can streamline its operations, minimize conflicts, and optimize resource utilization. The ability to generate reports on resource usage per week, month, and year provides valuable insights for management to make informed decisions regarding resource allocation and customer service improvements.

With the new system in place, the company will be able to efficiently manage its resources, improve customer satisfaction, and maximize revenue potential. The automated scheduling process will save time and effort for both customers and staff, resulting in a more seamless and efficient experience.

c) In conclusion, the development of the Scheduling Application has been a rewarding experience for our group. We have overcome various challenges, gained valuable knowledge, and created a system that addresses the coworking space company's scheduling needs. The benefits of the new system, including optimized resource utilization, improved customer service, and enhanced operational efficiency, are well within reach. By leveraging the power of technology and data-driven decision-making, the company is poised to thrive in the dynamic and competitive coworking industry.

Overall, the project underscores the importance of effective resource management and the value of well-designed scheduling systems in optimizing operations and delivering exceptional customer experiences. With the successful implementation of the Scheduling Application, we are confident that the coworking space company will enjoy increased productivity, enhanced customer satisfaction, and sustained growth in the years to come.