Table of Contents

[ER Model 3](#_Toc157290154)

[Entities 4](#_Toc157290155)

[Data Dictionary: 5](#_Toc157290156)

[Account 5](#_Toc157290157)

[Announcement 6](#_Toc157290158)

[Booking 7](#_Toc157290159)

[BookSlot 8](#_Toc157290160)

[Condo 9](#_Toc157290161)

[CondoMgmt 10](#_Toc157290162)

[CondoUsefulContact 10](#_Toc157290163)

[ContactCat 11](#_Toc157290164)

[Facility 11](#_Toc157290165)

[FacTimeSlot 12](#_Toc157290166)

[Feedback 13](#_Toc157290167)

[FeedbkCat 14](#_Toc157290168)

[ItemCategory 14](#_Toc157290169)

[ItemPhoto 15](#_Toc157290170)

[ItemRelated 15](#_Toc157290171)

[Likes 16](#_Toc157290172)

[Message 16](#_Toc157290173)

[Owner 17](#_Toc157290174)

[Staff 18](#_Toc157290175)

[TempVehLabel 18](#_Toc157290176)

[Tenant 19](#_Toc157290177)

[UsefulContact 20](#_Toc157290178)

[Vehicle 21](#_Toc157290179)

[VehicleLabel 21](#_Toc157290180)

[SQL Statements 23](#_Toc157290181)

[Tables 65](#_Toc157290182)

[ACCOUNT TABLE 65](#_Toc157290183)

[ANNOUNCEMENT TABLE 65](#_Toc157290184)

[BOOKING TABLE 65](#_Toc157290185)

[BOOKSLOT TABLE 66](#_Toc157290186)

[CONDO TABLE 66](#_Toc157290187)

[CONDOMGMT TABLE 66](#_Toc157290188)

[CONDOUSEFULCONTACT TABLE 67](#_Toc157290189)

[CONTACTCAT TABLE 67](#_Toc157290190)

[FACILITY TABLE 67](#_Toc157290191)

[FACILITYTIMESLOT TABLE 68](#_Toc157290192)

[FEEDBACK Table 68](#_Toc157290193)

[FEEDBKCAT TABLE 68](#_Toc157290194)

[ITEMCATEGORY TABLE 69](#_Toc157290195)

[ITEMPHOTO 69](#_Toc157290196)

[ITEMRELATED TABLE 69](#_Toc157290197)

[STAFF TABLE 71](#_Toc157290198)

[TEMPVEHLABEL TABLE 71](#_Toc157290199)

[TENANT TABLE 71](#_Toc157290200)

[USEFULCONTACT TABLE 72](#_Toc157290201)

[VEHICLE TABLE 72](#_Toc157290202)

[VEHICLELABEL TABLE 72](#_Toc157290203)

[Queries 73](#_Toc157290204)

# ER Model

A diagram of a network

Description automatically generated

# Entities

S/N Relation  
1 Account (AccID, AccName, AccAddress, AccCtcNo, AccEmail, <CondoID>, <ApprovedBy>)  
2 Announcement (AnnID, AnnText, AnnStartDate, AnnEndDate, <CondoMgmtID>)  
3 Booking (BookingID, BookingDate, BookingStatus, <AccID>, <FacID, TimeSlotSN, SlotDate>)  
4 BookSlot(<FacID, TimeSlotSN>, SlotDate, SlotStatus)  
5 Condo (CondoID, CondoName, CondoAddress)  
6 CondoMgmt (<CondoMgmtID>, ContactPerson, CtcPersonMobile)  
7 CondoUsefulContact (<CondoID>, <UsefulCtcID>)  
8 ContactCat (CtcCatID, CtcCatDesc)  
9 Facility (FacID, FacName, Deposit, <CondoID>)  
10 FacTimeSlot (<FacID>, TimeSlotSN, SlotDesc)  
11 Feedback (FbkID, FbkDesc, FbkDateTime, FbkStatus, <ByAccID>, <FbkCatID>, <CondoMgmtID>)  
12 FeedbkCat (FbkCatID, FbkCatDesc)  
13 ItemCategory (ItemCatID, ItemCatDesc)  
14 ItemPhoto (<ItemID>, Photo)  
15 ItemRelated (<ItemID>, ItemDesc, ItemPrice, ItemStatus, SaleOrRent, <ItemCatID>)  
16 Likes ( <AccID>, <MessageID>)  
17 Message (MsgID, MsgText, MsgType, <PostedBy>, <ReplyTo>)  
18 Owner (<OwnerID>, OwnStartDate, <CheckedBy>)  
19 Staff (StaffID, StaffName, StaffContactNo, StaffDateJoined, StaffRole)  
20 TempVehLabel (<VehLblAppID>, TempStartDate, TempExpiryDate)  
21 Tenant (<TenantID>, ContractStartDate, ContractEndDate, <VerifiedBy>)  
22 UsefulContact (UsefulCtcID, UsefulCtcName, UsefulCtcDesc, UsefulCtcPhone, <CtcCatID>)  
23 Vehicle (VehicleNo, IUNo, Ownership, Make, Model)  
24 VehicleLabel (VehLblAppID, VehLblStatus, VehLblNum, <VehicleNo>,<AppliedBy>, <IssuedBy>)

# Data Dictionary:

## Account

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | AccID | Unique account ID | int | Primary Key | No |
| 2 | AccName | The name of the account holder | varchar(100) |  | No |
| 3 | AccAddress | The home address of the account holder | varchar(255) |  | No |
| 4 | AccCtcNo | The contact number of the account holder | char(8) |  | Yes |
| 5 | AccEmail | The email address of the account holder | varchar(100) |  | Yes |
| 6 | CondoID | The condo that the account is related to | int | Foreign Key referencing Condo(CondoID) | Yes |
| 7 | ApprovedBy | The staff ID who approves the account | char(9) | Foreign Key referencing Staff(StaffID) | Yes |

#### 

#### 

## Announcement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | AnnID | Unique announcement ID | int | Primary Key | No |
| 2 | AnnText | The text message of the announcement | varchar(500) |  | No |
| 3 | AnnStartDate | The date when the announcement starts | smalldatetime |  | No |
| 4 | AnnEndDate | The date when the announcement ends | smalldatetime | CHECK (AnnStartDate <= AnnEndDate) | Yes |
| 5 | CondoMgmtID | The account ID of the condo management | int | Foreign Key referencing CondoMgmt (CondoMgmtID) | No |

## Booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | BookingID | Unique booking ID | int | Primary Key | No |
| 2 | BookingDate | The date when the booking is made | smalldatetime |  | No |
| 3 | BookingStatus | The status of the booking | varchar(50) | CHECK (BookingStatus IN ('Pending Payment','Confirmed','Cancelled')) | No |
| 4 | AccID | The account ID of the person who makes the booking | int | Foreign Key  Referencing Account (AccID) | No |
| 5 | FacID | The facility ID that is being booked | int | Composite Foreign Key  Referencing Facility(FacID) | No |
| 6 | TimeSlotSN | The facility's time slot number | char(4) | Composite Foreign Key  Referencing  FacTimeSlot(TimeSlotSN) | No |
| 7 | SlotDate | The date when the facility time slot is booked | smalldatetime | Composite Foreign Key  Referencing  BookSlot(SlotDate)  CHECK (DATEDIFF(HOUR, BookingDate, SlotDate) >= 24) | No |

## BookSlot

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | FacID | Unique facility ID that has the time slot | int | Composite Primary Key + Foreign Key referencing Facility(FacID) | No |
| 2 | TimeSlotSN | The sequence number of a time slot | char(4) | Composite Primary Key + Foreign Key referencing FacTimeSlot(TimeSlotSN) | No |
| 3 | SlotDate | The date when the facility time slot can be booked | smalldatetime | Composite Primary Key | No |
| 4 | SlotStatus | The status of the slot (e.g., Available, Booked, Maintenance) | varchar(20) |  | No |

## Condo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | CondoID | Unique condo ID | int | Primary Key | No |
| 2 | CondoName | The name of the condo | varchar(100) |  | No |
| 3 | CondoAddress | The address of the condo | varchar(200) |  | No |

## CondoMgmt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | CondoMgmtID | The account ID of the company (or staff) managing a condo | int | Primary Key + Foreign Key referencing Account(AccID) | No |
| 2 | ContactPerson | The contact person of the condo management company | varchar(100) |  | No |
| 3 | CtcPersonMobile | The mobile number of the contact person | Char(8) |  | No |

## CondoUsefulContact

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | CondoID | Unique ID of a condo | int | Composite Primary Key  Foreign Key  References Condo(CondoID) | No |
| 2 | UsefulCtcID | Unique ID of a condo's useful contact | char(5) | Composite Primary Key  Foreign Key  References UsefulContact(UsefulCtcID) | No |

## ContactCat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | CtcCatID | Unique category ID of a useful contact | int | Primary Key | No |
| 2 | CtcCatDesc | The description of a useful contact category | varchar(100) |  | No |

## Facility

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | FacID | Unique facility ID | int | Primary Key | No |
| 2 | FacName | The name or description of the facility | varchar(100) |  | No |
| 3 | Deposit | The amount of deposit required to book the facility | smallmoney |  | Yes |
| 4 | CondoID | The condo ID associated with the facility | int | Foreign Key  References Condo(CondoID) | No |

## FacTimeSlot

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | FacID | Facility ID | int | Composite Primary Key  Foreign Key  References Facility(FacID) | No |
| 2 | TimeSlotSN | The sequence number of a time slot | char(4) | Composite Primary Key | No |
| 3 | SlotDesc | Description of the time slot | varchar(100) |  | No |

## Feedback

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | FbkID | Unique feedback ID | int | Primary Key | No |
| 2 | FbkDesc | Description of the feedback | varchar(255) |  | No |
| 3 | FbkDateTime | Date and time when the feedback was given | smalldatetime |  | No |
| 4 | FbkStatus | Status of the feedback | varchar(20) |  | No |
| 5 | ByAccID | Account ID of the person who gave the feedback | int | Foreign Key  References  Account(AccID) | No |
| 6 | FbkCatID | Feedback category ID | char(4) | Foreign Key  References FeedbkCat(FbkCatID) | No |
| 7 | CondoMgmtID | Account ID of the condo  management staff who handles the feedback | int | Foreign Key  References CondoMgmt(CondoMgmtID) | Yes |

## FeedbkCat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | FbkCatID | Unique feedback category ID | char(4) | Primary Key | No |
| 2 | FbkCatDesc | Description of the feedback category | varchar(50) |  | No |

## ItemCategory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | ItemCatID | Unique item category ID | int | Primary Key | No |
| 2 | ItemCatDesc | Description of the item category | varchar(50) |  | No |

## ItemPhoto

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | ItemID | Unique Item ID | int | Primary Key  Foreign Key  REFERENCE ItemRelated(ItemID) | No |
| 2 | Photo | Photo of the item | varchar(20) |  | No |

## ItemRelated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | ItemID | Item ID | int | Primary key + Foreign Key  REFERENCE  Message(MsgID) | No |
| 2 | ItemDesc | Description of the item | varchar(100) |  | No |
| 3 | ItemPrice | Price of the item | money |  | Yes |
| 4 | ItemStatus | Status of the item | varchar(50) |  | No |
| 5 | SaleOrRent | Indicates whether the item is for sale | char(4) |  | No |
| 6 | ItemCatID | Item category ID | int | Foreign Key REFERENCES ItemCategory(ItemCatID) | No |

## Likes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | AccID | Account ID | int | Composite Primary Key  Foreign key references Account(AccId) | No |
| 2 | MessageID | Message ID | int | Composite Primary Key  Foreign key references Message(MsgID) | No |

## Message

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | MsgID | Unique message ID | int | Primary Key | No |
| 2 | MsgText | The text of the message | varchar(255) |  | No |
| 3 | MsgType | The type of the message | varchar(50) |  | No |
| 4 | PostedBy | The account ID of the person who posted the message | int | Foreign Key REFERENCES Account(AccID) | No |
| 5 | ReplyTo | The message ID this message is a reply to (if any) | int | Foreign Key  REFERENCES Message(MsgID) | Yes |

## Owner

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | OwnerID | Unique owner ID | int | Primary Key,  Foreign Key REFERENCES Account(AccountID) | No |
| 2 | OwnStartDate | The start date of ownership | date |  | No |
| 3 | CheckedBy | The staff ID of the person who checked the owner | int | Foreign Key REFERENCES Condomgmt(CondomgmtID) | No |

## Staff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | StaffID | Unique staff ID | char(9) | Primary Key | No |
| 2 | StaffName | The name of the staff member | varchar(100) |  | No |
| 3 | StaffContactNo | The contact number of the staff member | varchar(20) |  | No |
| 4 | StaffDateJoined | The date the staff member joined | date |  | No |
| 5 | StaffRole | The role of the staff member | varchar(50) |  | Yes |

## TempVehLabel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | VehLblAppID | Unique temporary vehicle label ID | int | Primary Key + Foreign key referencing VehicleLabel(VehLblAppID) | No |
| 2 | TempStartDate | The start date of the temporary label | smalldatetime |  | No |
| 3 | TempExpiryDate | The expiry date of the temporary label | smalldatetime | CHECK (TempExpiryDate > TempStartDate) | No |

#### 

## Tenant

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | TenantID | Unique tenant ID | int | Primary Key + Foreign key referencing Account (AccID) | No |
| 2 | ContractStartDate | The start date of the tenant's contract | smalldatetime |  | No |
| 3 | ContractEndDate | The end date of the tenant's contract | smalldatetime |  | No |
| 4 | VerifiedBy | The staff ID of the person who verified the tenant | int | Foreign Key referencing to condo management ID | No |

## UsefulContact

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | UsefulCtcID | Unique useful contact ID | Char(5) | Primary Key | No |
| 2 | UsefulCtcName | The name of the useful contact | varchar(100) |  | No |
| 3 | UsefulCtcDesc | Description of the useful contact | varchar(255) |  | No |
| 4 | UsefulCtcPhone | Phone number of the useful contact | char(8) |  | No |
| 5 | CtcCatID | The contact category ID that this useful contact belongs to | int | Foreign Key referencing to contactcat (CtcCatID) | No |

## Vehicle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | VehicleNo | Unique vehicle number | char(8) | Primary Key | No |
| 2 | IUNo | In-Unit Number for the vehicle | varchar(50) |  | No |
| 3 | Ownership | Ownership status of the vehicle | char(20) | Check if its “owned” ,“rented” ,“Company Car”,’ “Others” | No |
| 4 | Make | Make of the vehicle | varchar(50) |  | No |
| 5 | Model | Model of the vehicle | varchar(50) |  | No |

## VehicleLabel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Attribute** | **Description** | **Data Type** | **Constraints** | **Null Value** |
| 1 | VehLblAppID | Unique vehicle label application ID | int | Primary Key | No |
| 2 | VehLblStatus | Status of the vehicle label application | varchar(20) |  | No |
| 3 | VehLblNum | The vehicle label number assigned by the management | varchar(255) |  | No |
| 4 | VehicleNo | Vehicle number which the label is applied for | char(8) | Foreign Key referencing to Vehicle (Unique Vehicle Number) - VehicleNo | No |
| 5 | AppliedBy | Account ID of the person who makes the application | int | Foreign Key  Referencing to Account (account ID of person who made application) | No |
| 6 | IssuedBy | Staff ID of the person who issues the vehicle label | int | Foreign Key  Referencing to Account( account ID of condo management staff) | No |

--Instructions--

--1. Comment out everything below drop database mycondo and run the query--

--2. Comment out drop table mycondo and uncomment create database mycondo (but keep commenting out everything below create database mycondo) and run--

--3. Go to mycondo database--

--4. Comment out create and drop table mycondo and uncomment everything below (create and insert tables) and run--

--Assumptions--

--Only Singapore Carplates are accepted (char 8)

--Only Singapore phone numbers are accepted (char 8)

--drop database mycondo;

--create database mycondo;

CREATE TABLE ContactCat

(

CtcCatID INT NOT NULL,

CtcCatDesc VARCHAR(100) NOT NULL,

CONSTRAINT PK\_ContactCat PRIMARY KEY (CtcCatID)

);

INSERT INTO ContactCat

VALUES

(401, 'Management Office'),

(402, 'FnB'),

(403, 'Laundry'),

(404, 'Grocery Delivery Service'),

(405, 'Maintenance and Repair Service'),

(406, 'Fitness Center'),

(407, 'Healthcare'),

(408, 'Miscellaneous');

CREATE TABLE FeedbkCat

(

FbkCatID CHAR(4) NOT NULL,

FbkCatDesc VARCHAR(50) NOT NULL,

CONSTRAINT PK\_FeedbackCat PRIMARY KEY (FbkCatID)

);

INSERT INTO FeedbkCat

VALUES

('FC01', 'Maintenance'),

('FC02', 'Commends'),

('FC03', 'Complaints'),

('FC04', 'Cleanliness'),

('FC05', 'Security'),

('FC06', 'Plumbing'),

('FC07', 'Building Defects'),

('FC08', 'Parking'),

('FC09', 'Miscellaneous');

CREATE TABLE ItemCategory (

ItemCatID int NOT NULL,

ItemCatDesc varchar(50) NOT NULL

CONSTRAINT PK\_ItemCategory PRIMARY KEY (ItemCatID)

)

INSERT INTO ItemCategory (ItemCatID, ItemCatDesc) VALUES

(601, 'Homeware'),

(602, 'Fashion'),

(603, 'Furniture'),

(604, 'Electronics'),

(605, 'Books'),

(606, 'Toys and Games'),

(607, 'Sporting Goods'),

(608, 'Beauty and Personal Care'),

(609, 'Automotive'),

(610, 'Home Improvement'),

(611, 'Miscellaneous');

CREATE TABLE Staff

(

StaffID CHAR(9) NOT NULL,

StaffName VARCHAR(100) NOT NULL,

StaffContactNo VARCHAR(20) NOT NULL,

StaffDateJoined DATE NOT NULL,

StaffRole VARCHAR(50) NULL,

CONSTRAINT PK\_Staff PRIMARY KEY (StaffID)

);

INSERT INTO Staff (StaffID, StaffName, StaffContactNo, StaffDateJoined, StaffRole)

VALUES

('StaffID01', 'John Smith', '123-456-7890', '2023-01-15', 'Admin'),

('StaffID02', 'Jane Doe', '987-654-3210', '2023-02-20', 'Customer Service'),

('StaffID03', 'Michael Johnson', '555-123-4567', '2023-03-10', 'Tech Support'),

('StaffID04', 'Emily Wilson', '111-222-3333', '2023-04-05', 'Tech Support'),

('StaffID05', 'David Lee', '777-888-9999', '2023-05-12', 'Admin'),

('StaffID06', 'Sarah Turner', '444-555-6666', '2023-06-18', 'Admin'),

('StaffID07', 'Robert Jackson', '222-333-4444', '2023-07-22', 'Tech Support'),

('StaffID08', 'Linda White', '666-777-8888', '2023-08-30', 'Admin'),

('StaffID09', 'William Davis', '888-999-0000', '2023-09-05', 'Admin'),

('StaffID10', 'Michelle Hall', '333-444-5555', '2023-10-11', 'Customer Service');

CREATE TABLE Vehicle

(

VehicleNo CHAR(8) NOT NULL, --Assume only singapore carplates--

IUNo VARCHAR(50) NOT NULL,

Ownership VARCHAR(20) NOT NULL,

Make VARCHAR(50) NOT NULL,

Model VARCHAR(50) NOT NULL,

CONSTRAINT PK\_Vehicle PRIMARY KEY (VehicleNo),

CONSTRAINT CHK\_Ownership CHECK (Ownership IN ('Owned', 'Rented', 'Company Car', 'Others'))

);

INSERT INTO Vehicle (VehicleNo, IUNo, Ownership, Make, Model) VALUES

('SGA1234A', 'IU1234', 'Owned', 'Toyota', 'Corolla'),

('SGA2345B', 'IU2345', 'Rented', 'Honda', 'Civic'),

('SGA3456C', 'IU3456', 'Company Car', 'Ford', 'Focus'),

('SGA4567D', 'IU4567', 'Owned', 'Nissan', 'Sentra'),

('SGA5678E', 'IU5678', 'Rented', 'Chevrolet', 'Cruze'),

('SGA6789F', 'IU6789', 'Owned', 'Hyundai', 'Elantra'),

('SGA7890G', 'IU7890', 'Owned', 'Mazda', '3'),

('SGA8901H', 'IU8901', 'Rented', 'BMW', '3 Series'),

('SGA9012I', 'IU9012', 'Others', 'Audi', 'A4'),

('SGA0123J', 'IU0123', 'Owned', 'Mercedes', 'C-Class'),

('SGA1357K', 'IU1357', 'Owned', 'Tesla', 'Model S'),

('SGA2468L', 'IU2468', 'Rented', 'Kia', 'Sorento'),

('SGA3579M', 'IU3579', 'Owned', 'Subaru', 'Outback'),

('SGA4680N', 'IU4680', 'Owned', 'Jaguar', 'XE'),

('SGA5791O', 'IU5791', 'Rented', 'Lexus', 'RX'),

('SGA6802P', 'IU6802', 'Owned', 'Volvo', 'XC90'),

('SGA7913Q', 'IU7913', 'Owned', 'Porsche', 'Cayenne'),

('SGA8024R', 'IU8024', 'Rented', 'Volkswagen', 'Passat'),

('SGA9135S', 'IU9135', 'Others', 'Toyota', 'Prius'),

('SGA0246T', 'IU0246', 'Owned', 'Honda', 'Accord');

CREATE TABLE Condo

(

CondoID INT NOT NULL,

CondoName VARCHAR(100) NOT NULL,

CondoAddress VARCHAR(200) NOT NULL,

CONSTRAINT PK\_Condo PRIMARY KEY (CondoID)

);

INSERT INTO Condo

VALUES

(1, 'Sunset Heights Condominiums', '123 Sunset Heights Lane, Singapore'),

(2, 'Harborview Towers', '456 Harborview Avenue, Singapore'),

(3, 'Palm Grove Residences', '789 Palm Grove Street, Singapore'),

(4, 'Emerald Court Condos', '101 Emerald Court, Singapore'),

(5, 'Ocean Breeze Apartments', '222 Ocean Breeze Drive, Singapore'),

(6, 'Parkside Manor Condominiums', '333 Parkside Manor Road, Singapore'),

(7, 'Cityscape Plaza Residences', '444 Cityscape Plaza, Singapore'),

(8, 'Skyline Towers', '555 Skyline Towers Boulevard, Singapore'),

(9, 'Meadowview Estates', '666 Meadowview Estates Lane, Singapore'),

(10, 'Riverside Gardens Apartments', '777 Riverside Gardens Avenue, Singapore');

CREATE TABLE Account (

AccID INT NOT NULL,

AccName VARCHAR(100) NOT NULL,

AccAddress VARCHAR(255) NOT NULL,

AccCtcNo CHAR(8) NULL,

AccEmail VARCHAR(50) NULL,

CondoID INT NULL,

ApprovedBy CHAR(9) NULL,

CONSTRAINT PK\_Account PRIMARY KEY (AccID),

CONSTRAINT FK\_Account\_CondoID FOREIGN KEY (CondoID) REFERENCES Condo(CondoID),

CONSTRAINT FK\_Account\_ApprovedBy FOREIGN KEY (ApprovedBy) REFERENCES Staff(StaffID)

);

INSERT INTO Account (AccID, AccName, AccAddress, AccCtcNo, AccEmail, CondoID, ApprovedBy) VALUES

(1001, 'Sheng Xi Ling', '431 Clementi Ave 3 120431 #01-304', '67776690', 'shen\_g@gmail.com', 1, 'StaffID01'),

(1002, 'Chee Wei Le', '705 SIMS DRIVE 04-02 SHUN LI IND''L BLDG 87384', '68421846', 'weiwei@hotmail.com', 2, 'StaffID02'),

(1003, 'Say Yi Ming', '231B Victoria Street 188025', '63384752', 'saymyname@gmail.com', 3, 'StaffID03'),

(1004, 'Tung Jun Kai', '21 Tagore Lane S(787479)', '64598160', 'tungsten44@gmail.com', 4, 'StaffID04'),

(1005, 'Chang Cheng Hao', '14A Upper Cross Street 058330', '86945122', 'doublecHao@gmail.com', 5, 'StaffID05'),

(1006, 'Zhuang Xin Yi', 'Little India Shop Houses 20 Buffalo Road 219789', '62941421', 'xinyi123@yahoo.com', 6, 'StaffID06'),

(1007, 'Lau Zheng Jay', 'Juanda Haryono, Jurong Town 610005', '62256005', 'ZhengJay000@gmail.com', 7, 'StaffID07'),

(1008, 'Lau Zheng Min', 'Juanda Min, Jurong Town 610005', '62256015', 'ZhengMin000@gmail.com', 8, 'StaffID08'),

(1009, 'Lau Zheng Hay', 'Juanda Hay, Jurong Town 610005', '62256092', 'ZhengHay000@gmail.com', 9, 'StaffID09'),

(1010, 'Lau Zheng May', 'Juanda May, Jurong Town 610005', '62256055', 'ZhengMay000@gmail.com', 10, 'StaffID10'),

(1011, 'Sunset Heights Condominiums', '123 Sunset Heights Lane, Singapore', '12345678', 'sunsetheightscondo@gmail.com', 1, 'StaffID01'),

(1012, 'Harborview Towers', '456 Harborview Avenue, Singapore', '98765432', 'harborviewtowers@gmail.com', 2, 'StaffID02'),

(1013, 'Palm Grove Residences', '789 Palm Grove Street, Singapore', '87654321', 'palmgroveresidences@gmail.com', 3, 'StaffID03'),

(1014, 'Emerald Court Condos', '101 Emerald Court, Singapore', '98765432', 'emeraldcourtcondos@gmail.com', 4, 'StaffID04'),

(1015, 'Ocean Breeze Apartments', '222 Ocean Breeze Drive, Singapore', '87654321', 'oceanbreezeapartments@gmail.com', 5, 'StaffID01'),

(1016, 'Parkside Manor Condominiums', '333 Parkside Manor Road, Singapore', '98765432', 'parksidemanorcondos@gmail.com', 6, 'StaffID02'),

(1017, 'Cityscape Plaza Residences', '444 Cityscape Plaza, Singapore', '12345678', 'cityscapeplazaresidences@gmail.com', 7, 'StaffID03'),

(1018, 'Skyline Towers', '555 Skyline Towers Boulevard, Singapore', '23456789', 'skylinetowers@gmail.com', 8, 'StaffID04'),

(1019, 'Meadowview Estates', '666 Meadowview Estates Lane, Singapore', '34567890', 'meadowviewestates@gmail.com', 9, 'StaffID09'),

(1020, 'Riverside Gardens Apartments', '777 Riverside Gardens Avenue, Singapore', '45678901', 'riversidegardensapartments@gmail.com', 10, 'StaffID03'),

(1021, 'Sheng Xi Ling', '431 Clementi Ave 3 120431 #01-304', '67776690', 'shen\_g@gmail.com', 1, 'StaffID01'),

(1022, 'Chee Wei Le', '705 SIMS DRIVE 04-02 SHUN LI IND''L BLDG 87384', '68421846', 'weiwei@hotmail.com', 2, 'StaffID02'),

(1023, 'Say Yi Ming', '231B Victoria Street 188025', '63384752', 'saymyname@gmail.com', 3, 'StaffID03'),

(1024, 'Tung Jun Kai', '21 Tagore Lane S(787479)', '64598160', 'tungsten44@gmail.com', 4, 'StaffID04'),

(1025, 'Chang Cheng Hao', '14A Upper Cross Street 058330', '86945122', 'doublecHao@gmail.com', 5, 'StaffID05'),

(1026, 'Zhuang Xin Yi', 'Little India Shop Houses 20 Buffalo Road 219789', '62941421', 'xinyi123@yahoo.com', 6, 'StaffID06'),

(1027, 'Lau Zheng Jay', 'Juanda Haryono, Jurong Town 610005', '62256005', 'ZhengJay000@gmail.com', 7, 'StaffID07'),

(1028, 'Lau Zheng Min', 'Juanda Min, Jurong Town 610005', '62256015', 'ZhengMin000@gmail.com', 8, 'StaffID08'),

(1029, 'Lau Zheng Hay', 'Juanda Hay, Jurong Town 610005', '62256092', 'ZhengHay000@gmail.com', 9, 'StaffID09'),

(1030, 'Lau Zheng May', 'Juanda May, Jurong Town 610005', '62256055', 'ZhengMay000@gmail.com', 10, 'StaffID10'),

(1031, 'Jon Smith', '193/A3 Joo Chiat Road 427464', '12345678', 'Jonsmith123@hotmail.com', 1, 'StaffID01'),

(1032, 'Emily Johnson', '38 TAGORE LANE 110113', '98765432', 'JonEmily23@gmail.com', 2, 'StaffID02'),

(1033, 'Michael Lee', '20 Eng Hoon Street 312012', '87654321', 'michaelelee@hotmail.com', 3, 'StaffID03'),

(1034, 'Petty Smith', '#503, 12-159, Choa Chu Kang St. 51 321321', '98765432', 'petty.smith@example.com', 4, 'StaffID04'),

(1035, 'Alice Tan', '123 Orchard Road, Singapore', '87654321', 'alice.tan@example.com', 5, 'StaffID01'),

(1036, 'Bob Lim', '456 Marina Bay Sands, Singapore', '98765432', 'bob.lim@example.com', 6, 'StaffID02'),

(1037, 'Catherine Ng', '123 Chinatown Street, Singapore', '12345678', 'catherine.ng@example.com', 7, 'StaffID03'),

(1038, 'David Lee', '789 Little India, Singapore', '23456789', 'david.lee@example.com', 8, 'StaffID04'),

(1039, 'Eva Tan', '456 Sentosa Island, Singapore', '34567890', 'eva.tan@example.com', 9, 'StaffID09'),

(1040, 'Frank Lim', '101 Clarke Quay, Singapore', '45678901', 'frank.lim@example.com', 10, 'StaffID03');

CREATE TABLE CondoMgmt

(

CondoMgmtID INT NOT NULL,

ContactPerson VARCHAR(100) NOT NULL,

CtcPersonMobile char(8) NOT NULL,

CONSTRAINT PK\_CondoMgmt PRIMARY KEY (CondoMgmtID),

CONSTRAINT FK\_CondoMgmt\_AccID FOREIGN KEY (CondoMgmtID) REFERENCES Account(AccID)

);

INSERT INTO CondoMgmt (CondoMgmtID, ContactPerson, CtcPersonMobile) VALUES

(1011, 'Jon Smith', '12345678'),

(1012, 'Emily Johnson', '98765432'),

(1013, 'Michael Lee', '87654321'),

(1014, 'Petty Smith', '98765432'),

(1015, 'Alice Tan', '87654321'),

(1016, 'Bob Lim', '98765432'),

(1017, 'Catherine Ng', '12345678'),

(1018, 'David Lee', '23456789'),

(1019, 'Eva Tan', '34567890'),

(1020, 'Frank Lim', '45678901');

CREATE TABLE Announcement

(

AnnID INT NOT NULL,

AnnText VARCHAR(500) NOT NULL,

AnnStartDate smalldatetime NOT NULL,

AnnEndDate smalldatetime NULL,

CondoMgmtID INT NOT NULL,

CONSTRAINT Announcement\_PK PRIMARY KEY (AnnID),

CONSTRAINT FK\_Announcement\_CondoMgmtID FOREIGN KEY (CondoMgmtID) REFERENCES CondoMgmt(CondoMgmtID),

CONSTRAINT CHK\_Announcement\_Date CHECK (AnnStartDate <= AnnEndDate)

);

INSERT INTO Announcement (AnnID, AnnText, AnnStartDate, AnnEndDate, CondoMgmtID) VALUES

(10001, 'Community BBQ Event: Join us for a day of food, fun, and games at the condo courtyard!', '2020-10-12 08:00:00', '2020-10-12 22:59:59', 1011),

(10002, 'Pool Closure for Maintenance: The pool will be temporarily closed for cleaning and repairs from July 15th to July 19th. We apologize for any inconvenience caused.', '2020-07-15 08:00:00', '2020-07-19 22:59:59', 1012),

(10003, 'New Fitness Classes: Get fit and stay active with our latest lineup of fitness classes available at the condo gym. Check the schedule for details.', '2020-10-12 08:00:00', '2020-10-15 22:59:59', 1013),

(10004, 'Annual General Meeting: Mark your calendars for our upcoming AGM on October 5th. Join us to discuss important community matters and future plans.', '2020-10-12 08:00:00', '2020-10-12 22:59:59', 1014),

(10005, 'Parking Lot Resurfacing: Please be advised that sections of the condo parking lot will be under construction from August 10th to August 15th. Alternative parking arrangements have been made.', '2020-08-10 08:00:00', '2020-08-15 22:59:59', 1015),

(10006, 'Pet Adoption Drive: Help us find forever homes for adorable rescue animals! Visit the lobby on September 22nd to meet the furry friends in need of loving families.', '2020-09-22 08:00:00', '2020-09-22 22:59:59', 1016),

(10007, 'Fire Safety Drill: In compliance with safety regulations, a fire drill will be conducted on November 2nd at 10:00 AM. Please familiarize yourself with the evacuation procedures.', '2020-11-02 10:00:00', '2020-11-02 10:30:00', 1017),

(10008, 'Upcoming Social Mixer: Join your neighbors for a night of socializing and entertainment on December 8th. Its a great opportunity to meet new friends!', '2020-11-15 18:00:00', '2020-11-15 22:00:00', 1018),

(10009, 'Holiday Decor Contest: Participate in our annual holiday decor contest! Deck the halls and show off your festive spirit. Winners will be announced on December 20th.', '2020-11-20 08:00:00', '2020-12-20 22:59:59', 1019),

(10010, 'Winter Weather Advisory: Stay informed about winter weather conditions. In case of snow, please follow the snow removal guidelines provided by the condo management.', '2020-12-01 00:00:00', '2020-12-01 23:59:59', 1020),

(10011, 'Community Garden Workshop: Learn the basics of gardening and get involved in our community garden project. Workshop scheduled for January 10th, 2021.', '2020-12-10 08:00:00', '2021-01-10 22:59:59', 1016),

(10012, 'Condo Security Update: Enhanced security measures are being implemented. Residents are advised to update their access cards at the management office by January 5th, 2021.', '2020-12-15 08:00:00', '2021-01-05 22:59:59', 1020);

CREATE TABLE Tenant

(

TenantID INT NOT NULL,

ContractStartDate smalldatetime NOT NULL,

ContractEndDate smalldatetime NOT NULL,

VerifiedBy INT NOT NULL,

CONSTRAINT PK\_Tenant PRIMARY KEY (TenantID),

CONSTRAINT FK\_Tenant\_Account FOREIGN KEY (TenantID) REFERENCES Account(AccID),

CONSTRAINT FK\_Tenant\_VerifiedBy FOREIGN KEY (VerifiedBy) REFERENCES Condomgmt(CondomgmtID)

);

INSERT INTO Tenant (TenantID, ContractStartDate, ContractEndDate, VerifiedBy) VALUES

(1001, '2024-01-01', '2025-01-01', 1011),

(1002, '2024-02-01', '2025-02-01', 1012),

(1003, '2024-03-01', '2025-03-01', 1013),

(1004, '2024-04-01', '2025-04-01', 1014),

(1005, '2024-05-01', '2025-05-01', 1015),

(1006, '2024-06-01', '2025-06-01', 1016),

(1007, '2024-07-01', '2025-07-01', 1017),

(1008, '2024-08-01', '2025-08-01', 1018),

(1009, '2024-09-01', '2025-09-01', 1019),

(1010, '2024-10-01', '2025-10-01', 1020);

CREATE TABLE Message (

MsgID INT NOT NULL,

MsgText VARCHAR(255) NOT NULL,

MsgType VARCHAR(50) NOT NULL,

PostedBy INT NOT NULL,

ReplyTo INT NULL,

CONSTRAINT PK\_Message PRIMARY KEY (MsgID),

CONSTRAINT FK\_Message\_PostedBy FOREIGN KEY (PostedBy) REFERENCES Account(AccID),

CONSTRAINT FK\_Message\_ReplyTo FOREIGN KEY (ReplyTo) REFERENCES Message(MsgID)

);

INSERT INTO Message (MsgID, MsgText, MsgType, PostedBy, ReplyTo) VALUES

(1201, 'Anyone up for a movie night?', 'Chatterbox', 1001, NULL),

(1202, 'Selling my old bicycle, DM if interested!', 'Garage Sale', 1002, NULL),

(1203, 'Looking for a study buddy for programming. Any takers?', 'Find a Buddy', 1003, NULL),

(1204, 'Just ordered pizza for the team, who is in', 'Chatterbox', 1001, NULL),

(1205, 'Found a great deal on laptops, check it out!', 'Garage Sale', 1002, NULL),

(1206, 'Me!', 'Chatterbox', 1001, 1204),

(1207, 'Need recommendations for a good sci-fi book!', 'Chatterbox', 1003, NULL),

(1208, 'Offering tutoring services in math, contact me for details.', 'Find a Buddy', 1001, NULL),

(1209, 'Planning a hiking trip next weekend, join us!', 'Chatterbox', 1002, NULL),

(1210, 'Renting camera, message for details.', 'Garage Sale', 1003, NULL),

(1211, 'Does anyone know a good place for brunch in the city?', 'Chatterbox', 1001, NULL),

(1212, 'Selling a headphones, message for details.', 'Garage Sale', 1005, NULL),

(1213, 'Selling gaming computer, used for 1 year, message for details.', 'Garage Sale', 1024, NULL),

(1214, 'Selling new airpods pro, message for details.', 'Garage Sale', 1026, NULL),

(1215, 'Selling PS5, message for details.', 'Garage Sale', 1028, NULL),

(1216, 'Selling gaming laptop, message for details.', 'Garage Sale', 1022, NULL),

(1217, 'Selling Fitness Tracker, message for details.', 'Garage Sale', 1036, NULL),

(1218, 'Renting mountain bike, message for details.', 'Garage Sale', 1034, NULL);

CREATE TABLE ItemRelated (

ItemID INT NOT NULL,

ItemDesc VARCHAR(100) NOT NULL,

ItemPrice MONEY NULL,

ItemStatus VARCHAR(50) NOT NULL,

SaleOrRent CHAR(4) NOT NULL,

ItemCatID INT NOT NULL,

CONSTRAINT PK\_ItemRelated PRIMARY KEY (ItemID),

CONSTRAINT FK\_ItemRelated\_MsgID FOREIGN KEY (ItemID) REFERENCES Message(MsgID),

CONSTRAINT FK\_ItemRelated\_ItemCatID FOREIGN KEY (ItemCatID) REFERENCES ItemCategory(ItemCatID),

CONSTRAINT CHK\_ItemRelated CHECK (SaleOrRent IN ('Sale', 'Rent')),

);

INSERT INTO ItemRelated (ItemID, ItemDesc, ItemPrice, ItemStatus, SaleOrRent, ItemCatID) VALUES

(1202, 'Old Bicycle', 199.99, 'Available', 'Sale', 607),

(1205, 'Laptop', 1299.99, 'Sold', 'Sale', 604),

(1210, 'Camera', 49.99, 'Available', 'Rent', 604),

(1212, 'Headphones', 79.99, 'Available', 'Sale', 604),

(1213, 'Gaming Computer', 899.99, 'Sold', 'Sale', 604),

(1214, 'Airpods Pro', 199.99, 'Available', 'Sale', 604),

(1215, 'PS5', 550.00, 'Available', 'Sale', 604),

(1216, 'Gaming Laptop', 89.99, 'Sold', 'Rent', 604),

(1217, 'Fitness Tracker', 69.99, 'Available', 'Sale', 607),

(1218, 'Mountain Bike', 100.00, 'Available', 'Rent', 607);

CREATE TABLE ItemPhoto (

ItemID INT NOT NULL,

Photo VARCHAR(20) NOT NULL,

CONSTRAINT PK\_ItemPhoto PRIMARY KEY (ItemID, Photo),

CONSTRAINT FK\_ItemPhoto FOREIGN KEY (ItemID) REFERENCES ItemRelated(ItemID)

);

INSERT INTO ItemPhoto (ItemID, Photo) VALUES

(1202, 'Photo 1'),

(1205, 'Photo 2'),

(1210, 'Photo 3'),

(1212, 'Photo 4'),

(1213, 'Photo 5'),

(1214, 'Photo 6'),

(1215, 'Photo 7'),

(1216, 'Photo 8'),

(1217, 'Photo 9'),

(1218, 'Photo 10');

CREATE TABLE Facility (

FacID INT NOT NULL,

FacName VARCHAR(100) NOT NULL,

Deposit SMALLMONEY NULL,

CondoID INT NOT NULL,

CONSTRAINT PK\_Facility PRIMARY KEY (FacID),

CONSTRAINT FK\_Facility\_Condo FOREIGN KEY (CondoID) REFERENCES Condo(CondoID)

);

INSERT INTO Facility (FacID, FacName, Deposit, CondoID) VALUES

(301, 'Function Hall', 50.00, 1),

(302, 'Sports Court', NULL, 1),

(303, 'Event Room', 10.00, 1),

(304, 'Sports Court', NULL, 2),

(305, 'Childcare Services', 5.00, 2),

(306, 'Sports Court', NULL, 3),

(307, 'Outdoor Grill', 2.00, 3),

(308, 'Sports Court', NULL, 4),

(309, 'Sports Court', NULL, 5),

(310, 'Event Room', 10.00, 5);

CREATE TABLE FacTimeSlot (

FacID INT NOT NULL,

TimeSlotSN CHAR(4) NOT NULL,

SlotDesc VARCHAR(100) NOT NULL,

CONSTRAINT PK\_FacTimeSlot PRIMARY KEY (FacID, TimeSlotSN),

CONSTRAINT FK\_FacTimeSlot\_Facility FOREIGN KEY (FacID) REFERENCES Facility(FacID)

);

INSERT INTO FacTimeSlot (FacID, TimeSlotSN, SlotDesc) VALUES

(301, 'SN01', '8am-10am'),

(301, 'SN02', '10am-12pm'),

(301, 'SN03', '12pm-2pm'),

(302, 'SN01', '8am-10am'),

(302, 'SN02', '10am-12pm'),

(302, 'SN03', '12pm-2pm'),

(303, 'SN01', '8am-10am'),

(303, 'SN02', '10am-12pm'),

(303, 'SN03', '12pm-2pm'),

(304, 'SN01', '8am-10am'),

(304, 'SN02', '10am-12pm'),

(304, 'SN03', '12pm-2pm'),

(305, 'SN01', '8am-10am'),

(305, 'SN02', '10am-12pm'),

(305, 'SN03', '12pm-2pm'),

(306, 'SN01', '8am-10am'),

(306, 'SN02', '10am-12pm'),

(306, 'SN03', '12pm-2pm'),

(307, 'SN01', '8am-10am'),

(307, 'SN02', '10am-12pm'),

(307, 'SN03', '12pm-2pm'),

(308, 'SN01', '8am-10am'),

(308, 'SN02', '10am-12pm'),

(308, 'SN03', '12pm-2pm'),

(309, 'SN01', '8am-10am'),

(309, 'SN02', '10am-12pm'),

(309, 'SN03', '12pm-2pm'),

(310, 'SN01', '8am-10am'),

(310, 'SN02', '10am-12pm'),

(310, 'SN03', '12pm-2pm');

CREATE TABLE Feedback (

FbkID INT NOT NULL,

FbkDesc VARCHAR(255) NOT NULL,

Fbkdatetime smalldatetime NOT NULL,

FbkStatus VARCHAR(20) NOT NULL,

ByAccID INT NOT NULL,

FbkCatID CHAR(4) NOT NULL,

CondoMgmtID INT NULL,

CONSTRAINT PK\_Feedback PRIMARY KEY (FbkID),

CONSTRAINT FK\_Feedback\_ByAccID FOREIGN KEY (ByAccID) REFERENCES Account(AccID),

CONSTRAINT FK\_Feedback\_FbkCatID FOREIGN KEY (FbkCatID) REFERENCES FeedbkCat(FbkCatID),

CONSTRAINT FK\_Feedback\_CondoMgmtID FOREIGN KEY (CondoMgmtID) REFERENCES CondoMgmt(CondoMgmtID)

);

INSERT INTO Feedback (FbkID, FbkDesc, FbkDateTime, FbkStatus, ByAccID, FbkCatID, CondoMgmtID) VALUES

(501, 'Elevator maintenance required', '2024-12-29 08:30:00', 'Open', 1001, 'FC01', NULL),

(502, 'Great service by the front desk', '2024-01-01 09:15:00', 'Attended', 1002, 'FC02', 1011),

(503, 'Noise complaint in the night', '2024-01-04 00:30:00', 'In Progress', 1023, 'FC03', 1012),

(504, 'Faulty light fixtures in hallway', '2024-01-05 19:00:00', 'Attended', 1004, 'FC01', 1013),

(505, 'Request for additional security at night', '2024-01-09 20:05:00', 'In Progress', 1005, 'FC05', 1011),

(506, 'Pool cleaning needed', '2024-01-10 20:00:00', 'In Progress', 1022, 'FC04', 1014),

(507, 'Leaking pipes in the basement area', '2024-01-11 08:45:00', 'Open', 1003, 'FC06', NULL),

(508, 'Cracks in the exterior wall of building B', '2024-01-13 12:10:00', 'In Progress', 1028, 'FC07', 1016),

(509, 'Rusty water pipes in building A', '2024-01-15 10:10:00', 'In Progress', 1007, 'FC06', 1018),

(510, 'Parking space shortage during weekends', '2024-01-16 10:20:00', 'Open', 1008, 'FC08', NULL);

CREATE TABLE Owner (

OwnerID INT NOT NULL,

OwnStartDate date NOT NULL,

CheckedBy INT NOT NULL,

CONSTRAINT PK\_Owner PRIMARY KEY (OwnerID),

CONSTRAINT FK\_Owner\_OwnerID FOREIGN KEY (OwnerID) REFERENCES Account(AccID),

CONSTRAINT FK\_Owner\_CheckedBy FOREIGN KEY (CheckedBy) REFERENCES CondoMgmt(CondoMgmtID)

);

INSERT INTO Owner (OwnerID, OwnStartDate, CheckedBy) VALUES

(1021, '2022-01-15', 1014),

(1022, '2021-09-05', 1018),

(1023, '2023-03-22', 1013),

(1024, '2020-11-10', 1011),

(1025, '2022-07-18', 1016),

(1026, '2021-04-03', 1011),

(1027, '2023-01-08', 1013),

(1028, '2020-08-27', 1020),

(1029, '2022-12-14', 1017),

(1030, '2021-06-29', 1014),

(1031, '2022-05-20', 1012),

(1032, '2021-11-08', 1015),

(1033, '2023-02-17', 1018),

(1034, '2020-09-30', 1016),

(1035, '2022-08-14', 1011),

(1036, '2021-02-25', 1017),

(1037, '2023-05-12', 1013),

(1038, '2020-12-03', 1019),

(1039, '2022-10-19', 1014),

(1040, '2021-07-06', 1020);

CREATE TABLE Likes (

AccID INT NOT NULL,

MessageID INT NOT NULL,

CONSTRAINT PK\_Likes PRIMARY KEY (AccID,MessageID),

CONSTRAINT FK\_Likes\_AccID FOREIGN KEY (AccID) REFERENCES Account(AccID),

CONSTRAINT FK\_Likes\_MsgID FOREIGN KEY (MessageID) REFERENCES Message(MsgID)

);

INSERT INTO Likes (AccID, MessageID) VALUES

(1001, 1201),

(1002, 1202),

(1003, 1203),

(1004, 1201),

(1005, 1205),

(1006, 1204),

(1007, 1202),

(1008, 1203),

(1009, 1206),

(1010, 1205);

CREATE TABLE VehicleLabel

(

VehLblAppID int NOT NULL,

VehLblStatus varchar(20) NOT NULL,

VehLblNum varchar(255) NOT NULL,

VehicleNo char(8) NOT NULL,

AppliedBy int NOT NULL,

IssuedBy int NOT NULL,

CONSTRAINT PK\_VehicleLabel PRIMARY KEY (VehLblAppID),

CONSTRAINT FK\_VehicleLabel\_Vehicle FOREIGN KEY (VehicleNo) REFERENCES Vehicle(VehicleNo),

CONSTRAINT FK\_VehicleLabel\_AppliedBy FOREIGN KEY (AppliedBy) REFERENCES Account(AccID),

CONSTRAINT FK\_VehicleLabel\_IssuedBy FOREIGN KEY (IssuedBy) REFERENCES CondoMgmt(CondoMgmtID)

);

INSERT INTO VehicleLabel (VehLblAppID, VehLblStatus, VehLblNum, VehicleNo, AppliedBy, IssuedBy)

VALUES

(100001, 'Approved', 'LBL001', 'SGA1234A', 1002, 1011),

(100002, 'Pending', 'LBL002', 'SGA2345B', 1003, 1012),

(100003, 'Rejected', 'LBL003', 'SGA3456C', 1004, 1013),

(100004, 'Approved', 'LBL004', 'SGA4567D', 1005, 1014),

(100005, 'Pending', 'LBL005', 'SGA5678E', 1006, 1015),

(100006, 'Approved', 'LBL006', 'SGA6789F', 1007, 1016),

(100007, 'Rejected', 'LBL007', 'SGA7890G', 1008, 1017),

(100008, 'Approved', 'LBL008', 'SGA8901H', 1009, 1018),

(100009, 'Pending', 'LBL009', 'SGA9012I', 1021, 1019),

(100010, 'Approved', 'LBL010', 'SGA0123J', 1022, 1020),

(100011, 'Approved', 'LBL011', 'SGA1357K', 1023,1011),

(100012, 'Pending', 'LBL012', 'SGA2468L', 1024, 1012),

(100013, 'Rejected', 'LBL013', 'SGA3579M', 1025, 1013),

(100014, 'Approved', 'LBL014', 'SGA4680N', 1026, 1014),

(100015, 'Pending', 'LBL015', 'SGA5791O', 1027, 1015),

(100016, 'Approved', 'LBL016', 'SGA6802P', 1028, 1016),

(100017, 'Rejected', 'LBL017', 'SGA7913Q', 1029, 1017),

(100018, 'Approved', 'LBL018', 'SGA8024R', 1030, 1018),

(100019, 'Pending', 'LBL019', 'SGA9135S', 1031, 1019),

(100020, 'Approved', 'LBL020', 'SGA0246T', 1032, 1020);

CREATE TABLE TempVehLabel

(

VehLblAppID int NOT NULL,

TempStartDate smalldatetime NOT NULL,

TempExpiryDate smalldatetime NOT NULL,

CONSTRAINT PK\_TempVehLabel PRIMARY KEY (VehLblAppID),

CONSTRAINT FK\_TempVehLabel\_VehicleLabel FOREIGN KEY (VehLblAppID) REFERENCES VehicleLabel(VehLblAppID),

CONSTRAINT CHK\_TempVehLabel CHECK (TempExpiryDate > TempStartDate)

);

INSERT INTO TempVehLabel (VehLblAppID, TempStartDate, TempExpiryDate) VALUES

(100001, '2024-01-01', '2024-01-31'),

(100002, '2024-02-01', '2024-02-28'),

(100003, '2024-03-01', '2024-03-31'),

(100004, '2024-04-01', '2024-04-30'),

(100005, '2024-05-01', '2024-05-31'),

(100006, '2024-06-01', '2024-06-30'),

(100007, '2024-07-01', '2024-07-31'),

(100008, '2024-08-01', '2024-08-31'),

(100009, '2024-09-01', '2024-09-30'),

(100010, '2024-10-01', '2024-10-31');

CREATE TABLE UsefulContact

(

UsefulCtcID char(5) NOT NULL,

UsefulCtcName varchar(100) NOT NULL,

UsefulCtcDesc varchar(255) NULL,

UsefulCtcPhone char(8) NOT NULL,

CtcCatID int NOT NULL,

CONSTRAINT PK\_UsefulContact PRIMARY KEY (UsefulCtcID),

CONSTRAINT FK\_UsefulContact\_CtcCatID FOREIGN KEY (CtcCatID) REFERENCES ContactCat(CtcCatID)

);

INSERT INTO UsefulContact (UsefulCtcID, UsefulCtcName, UsefulCtcDesc, UsefulCtcPhone, CtcCatID) VALUES

('UC001', 'ABC Plumbing', 'Plumbing services', '12345670', 405),

('UC002', 'XYZ Groceries', 'Local grocery store', '23456781', 404),

('UC003', 'Quick Cleaners', 'Cleaning services', '34567892', 403),

('UC004', 'Happy Pets', 'Pet care services', '45678903', 408),

('UC005', 'Tech Solutions', 'Technology repair services', '56789014', 405),

('UC006', 'Green Gardens', 'Landscaping services', '67890125', 405),

('UC007', 'Secure Homes', 'Home security systems', '78901236', 408),

('UC008', 'Fitness Plus', 'Gym and fitness center', '89012347', 406),

('UC009', 'Book Haven', 'Bookstore', '90123458', 407),

('UC010', 'Cafe Delight', 'Local cafe', '01234589', 402);

CREATE TABLE CondoUsefulContact

(

CondoID int NOT NULL,

UsefulCtcID char(5) NOT NULL,

CONSTRAINT PK\_CondoUsefulContact PRIMARY KEY (CondoID, UsefulCtcID),

CONSTRAINT FK\_CondoUsefulContact\_CondoID FOREIGN KEY (CondoID) REFERENCES Condo(CondoID),

CONSTRAINT FK\_CondoUsefulContact\_UsefulCtcID FOREIGN KEY (UsefulCtcID) REFERENCES UsefulContact(UsefulCtcID)

);

INSERT INTO CondoUsefulContact (CondoID, UsefulCtcID) VALUES

(1, 'UC001'),

(1, 'UC002'),

(1, 'UC003'),

(2, 'UC004'),

(2, 'UC005'),

(3, 'UC006'),

(3, 'UC007'),

(4, 'UC008'),

(4, 'UC009'),

(5, 'UC010');

CREATE TABLE BookSlot

(

FacID int NOT NULL,

TimeSlotSN char(4) NOT NULL,

SlotDate smalldatetime NOT NULL,

SlotSatus varchar(20) NOT NULL,

CONSTRAINT PK\_BookSlot PRIMARY KEY (FacID,TimeSlotSN,SlotDate),

CONSTRAINT FK\_BookSlot Foreign Key (FacID, TimeSLotSN) references FacTimeSlot(FacID, TimeSlotSN)

);

INSERT INTO BookSlot

VALUES

(301, 'SN01', '2022-02-20 08:00:00', 'Available'),

(301, 'SN02', '2022-02-20 10:00:00', 'Booked'),

(301, 'SN03', '2022-02-20 12:00:00 ', 'Available'),

(301, 'SN01', '2022-02-26 08:00:00', 'Available'),

(301, 'SN02', '2022-02-26 10:00:00', 'Available'),

(301, 'SN03', '2022-02-26 12:00:00', 'Available'),

(302, 'SN01', '2022-02-21 08:00:00', 'Booked'),

(302, 'SN02', '2022-02-21 10:00:00', 'Available'),

(302, 'SN03', '2022-02-21 12:00:00', 'Booked'),

(302, 'SN01', '2022-02-26 08:00:00', 'Booked'),

(302, 'SN02', '2022-02-26 10:00:00', 'Available'),

(302, 'SN03', '2022-02-26 12:00:00', 'Booked'),

(303, 'SN01', '2022-02-24 08:00:00', 'Available'),

(303, 'SN02', '2022-02-24 10:00:00', 'Booked'),

(303, 'SN03', '2022-02-24 12:00:00', 'Available'),

(303, 'SN01', '2022-02-26 08:00:00', 'Available'),

(303, 'SN02', '2022-02-26 10:00:00', 'Booked'),

(303, 'SN03', '2022-02-26 12:00:00', 'Available'),

(304, 'SN01', '2022-02-23 08:00:00', 'Available'),

(304, 'SN02', '2022-02-23 10:00:00', 'Available'),

(304, 'SN03', '2022-02-23 12:00:00', 'Booked'),

(304, 'SN01', '2022-02-26 08:00:00', 'Available'),

(304, 'SN02', '2022-02-26 10:00:00', 'Available'),

(304, 'SN03', '2022-02-26 12:00:00', 'Booked'),

(305, 'SN01', '2022-02-23 08:00:00', 'Booked'),

(305, 'SN02', '2022-02-23 10:00:00', 'Booked'),

(305, 'SN03', '2022-02-23 12:00:00', 'Available'),

(305, 'SN01', '2022-02-26 08:00:00', 'Booked'),

(305, 'SN02', '2022-02-26 10:00:00', 'Booked'),

(305, 'SN03', '2022-02-26 12:00:00', 'Available'),

(306, 'SN01', '2022-02-24 08:00:00', 'Available'),

(306, 'SN02', '2022-02-24 10:00:00', 'Available'),

(306, 'SN03', '2022-02-24 12:00:00', 'Booked'),

(306, 'SN01', '2022-02-26 08:00:00', 'Available'),

(306, 'SN02', '2022-02-26 10:00:00', 'Available'),

(306, 'SN03', '2022-02-26 12:00:00', 'Booked'),

(307, 'SN01', '2022-02-24 08:00:00', 'Booked'),

(307, 'SN02', '2022-02-24 10:00:00', 'Available'),

(307, 'SN03', '2022-02-24 12:00:00', 'Available'),

(307, 'SN01', '2022-02-26 08:00:00', 'Booked'),

(307, 'SN02', '2022-02-26 10:00:00', 'Available'),

(307, 'SN03', '2022-02-26 12:00:00', 'Available'),

(308, 'SN01', '2022-02-25 08:00:00', 'Available'),

(308, 'SN02', '2022-02-25 10:00:00', 'Booked'),

(308, 'SN03', '2022-02-25 12:00:00', 'Available'),

(308, 'SN01', '2022-02-26 08:00:00', 'Available'),

(308, 'SN02', '2022-02-26 10:00:00', 'Booked'),

(308, 'SN03', '2022-02-26 12:00:00', 'Available'),

(309, 'SN01', '2022-02-25 08:00:00', 'Available'),

(309, 'SN02', '2022-02-25 10:00:00', 'Available'),

(309, 'SN03', '2022-02-25 12:00:00', 'Booked'),

(309, 'SN01', '2022-02-26 08:00:00', 'Available'),

(309, 'SN02', '2022-02-26 10:00:00', 'Available'),

(309, 'SN03', '2022-02-26 12:00:00', 'Booked'),

(310, 'SN01', '2022-02-27 08:00:00', 'Maintenance'),

(310, 'SN02', '2022-02-27 10:00:00', 'Maintenance'),

(310, 'SN03', '2022-02-27 12:00:00', 'Maintenance');

CREATE TABLE Booking

(

BookingID int NOT NULL,

BookingDate smalldatetime NOT NULL,

BookingStatus varchar(50) NOT NULL CHECK (BookingStatus IN ('Pending Payment','Confirmed','Cancelled')),

AccID int NOT NULL,

FacID int NOT NULL,

TimeSlotSN char(4) NOT NULL,

SlotDate smalldatetime NOT NULL,

CONSTRAINT PK\_Booking PRIMARY KEY (BookingID),

CONSTRAINT FK\_Booking\_AccID FOREIGN KEY (AccID) REFERENCES Account(AccID),

CONSTRAINT FK\_Booking\_FacID\_TimeSlot FOREIGN KEY (FacID, TimeSlotSN, SlotDate) REFERENCES BookSlot(FacID, TimeSlotSN, SlotDate),

CONSTRAINT CK\_Booking CHECK (DATEDIFF(HOUR, BookingDate, SlotDate) >= 24) --checkpoint said booking needs to be made at least 24 hour in advance--

);

INSERT INTO Booking

VALUES

(101, '2022-02-15 08:30:00', 'Pending Payment', 1001, 301, 'SN02', '2022-02-20 10:00:00'),

(102, '2022-02-15 09:15:00', 'Confirmed', 1002, 302, 'SN03', '2022-02-21 12:00:00'),

(103, '2022-02-15 10:30:00', 'Cancelled', 1002, 302, 'SN01', '2022-02-21 08:00:00'),

(104, '2022-02-15 11:45:00', 'Confirmed', 1003, 303, 'SN02', '2022-02-24 10:00:00'),

(105, '2022-02-15 13:15:00', 'Pending Payment', 1003, 304, 'SN03', '2022-02-23 12:00:00'),

(106, '2022-02-15 15:30:00', 'Confirmed', 1004, 305, 'SN01', '2022-02-26 08:00:00'),

(107, '2022-02-16 16:45:00', 'Pending Payment', 1004, 305, 'SN02', '2022-02-26 10:00:00'),

(108, '2022-02-17 11:30:00', 'Confirmed', 1005, 306, 'SN03', '2022-02-24 12:00:00'),

(109, '2022-02-17 12:45:00', 'Cancelled', 1005, 307, 'SN01', '2022-02-24 08:00:00'),

(110, '2022-02-17 14:15:00', 'Confirmed', 1006, 308, 'SN02', '2022-02-25 10:00:00'),

(111, '2022-02-17 16:30:00', 'Cancelled', 1006, 309, 'SN03', '2022-02-25 12:00:00'),

(112, '2022-02-18 09:45:00', 'Pending Payment', 1007, 301, 'SN02', '2022-02-26 10:00:00'),

(113, '2022-02-18 11:30:00', 'Cancelled', 1008, 302, 'SN01', '2022-02-26 08:00:00'),

(114, '2022-02-19 09:00:00', 'Confirmed', 1009, 303, 'SN02', '2022-02-26 10:00:00'),

(115, '2022-02-19 09:15:00', 'Pending Payment', 1003, 304, 'SN03', '2022-02-26 12:00:00'),

(116, '2022-02-19 09:30:00', 'Confirmed', 1004, 305, 'SN01', '2022-02-26 08:00:00'),

(117, '2022-02-19 09:45:00', 'Pending Payment', 1004, 305, 'SN02', '2022-02-26 10:00:00'),

(118, '2022-02-20 10:00:00', 'Confirmed', 1005, 306, 'SN03', '2022-02-26 12:00:00'),

(119, '2022-02-21 10:15:00', 'Cancelled', 1005, 307, 'SN01', '2022-02-26 08:00:00'),

(120, '2022-02-22 10:30:00', 'Confirmed', 1006, 308, 'SN02', '2022-02-26 10:00:00'),

(121, '2022-02-23 10:45:00', 'Cancelled', 1006, 309, 'SN03', '2022-02-26 12:00:00'),

(122, '2022-02-23 11:00:00', 'Confirmed', 1002, 302, 'SN03', '2022-02-26 12:00:00');

# Tables

## ACCOUNT TABLE

## A screenshot of a computer Description automatically generated

## ANNOUNCEMENT TABLE

## A screenshot of a calendar Description automatically generated

## BOOKING TABLE

## A screenshot of a computer Description automatically generated

## BOOKSLOT TABLE

## A table with numbers and letters Description automatically generated

## A screenshot of a computer Description automatically generated

## CONDO TABLE

## A screenshot of a computer Description automatically generated

## CONDOMGMT TABLE

## A screenshot of a computer Description automatically generated

## CONDOUSEFULCONTACT TABLE

## A screenshot of a computer Description automatically generated

## CONTACTCAT TABLE

## A screenshot of a computer Description automatically generated

## 

## FACILITY TABLE

## A screenshot of a table Description automatically generated

## FACILITYTIMESLOT TABLE

## A screenshot of a computer Description automatically generated

## 

## FEEDBACK Table

## A screenshot of a computer Description automatically generated

## FEEDBKCAT TABLE

## A screenshot of a computer Description automatically generated

## 

## ITEMCATEGORY TABLE

A screenshot of a computer

Description automatically generated

## ITEMPHOTO

## A screenshot of a computer Description automatically generated

## ITEMRELATED TABLE

## A screenshot of a computer Description automatically generated

## LIKES TABLE

A screenshot of a computer

Description automatically generated

## MESSAGE TABLE

A white text box with black text

Description automatically generated

## OWNER TABLE

## A screenshot of a table Description automatically generated

## STAFF TABLE

## A screenshot of a phone number Description automatically generated

## TEMPVEHLABEL TABLE

## A screenshot of a computer Description automatically generated

## TENANT TABLE

A screenshot of a computer screen

Description automatically generated

## USEFULCONTACT TABLE

A list of services

Description automatically generated

## VEHICLE TABLE

## A screenshot of a computer Description automatically generated

## VEHICLELABEL TABLE

## A screenshot of a computer Description automatically generated

# 

# Queries

DYLAN TEO

-- 1. List the number of owners and tenants and the amount of bookings made in each condo --

select c.CondoID, c.CondoName,count(distinct o.ownerid) as NumberOfOwners, count(distinct t.tenantid) as NumberOfTenants, COUNT(distinct b.bookingid) as TotalBookings

from condo c

left join

(select a.CondoID, o.OwnerID

from account a

inner join owner o on a.AccID = o.OwnerID

group by a.CondoID, o.OwnerID)

as o on c.CondoID = o.CondoID

left join

(select a.CondoID, t.TenantID

from account a

inner join tenant t on a.AccID = t.TenantID

group by a.CondoID, t.TenantID)

as t on c.CondoID = t.CondoID

left join

(select c.CondoID, b.BookingID

from condo c

inner join

facility f on c.CondoID = f.CondoID

inner join

booking b on f.FacID = b.FacID)

as b on b.CondoID = c.CondoID

group by c.CondoID, c.CondoName;

-- 2. Find the total value of items listed for sale in each condo, in descending order --

SELECT c.CondoID, c.CondoName, SUM(ir.ItemPrice) AS TotalValueOfItems

from Condo c

left join

Account a on a.CondoID = c.CondoID

left join

message m on m.PostedBy = a.AccID

left join

ItemRelated ir on ir.ItemID = m.MsgID

group by c.CondoID, c.CondoName

order by TotalValueOfItems desc;

-- 3. List the amount of times each condomgmt company has interacted with the condo using the app (Announcements, Bookings, Feedback Attended/Completed) --

select c.CondoID,count(distinct an.AnnID) as AnnouncementsMade, count(distinct f.FbkID) as 'FeedbacksAttended/Completed', count(distinct b.BookingID) as BookingsConfirmed, count(distinct an.AnnID) + count(distinct f.FbkID) + count(distinct b.BookingID) as NoOfActivity

from condo c

left join

(select a.CondoID, f.FbkID

from account a

inner join CondoMgmt cm on cm.CondoMgmtID = a.AccID

inner join Feedback f on f.CondoMgmtID = cm.CondoMgmtID

where f.FbkStatus in ('In Progress', 'Attended')

group by a.CondoID, f.FbkID)

as f on f.CondoID = c.CondoID

left join

(select a.CondoID, an.AnnID

from account a

inner join CondoMgmt cm on cm.CondoMgmtID = a.AccID

inner join Announcement an on an.CondoMgmtID = cm.CondoMgmtID

group by a.CondoID, an.AnnID)

as an on an.CondoID = c.CondoID

left join

(select f.CondoID, b.BookingID

from facility f

inner join booking b on b.FacID = f.FacID

where b.BookingStatus = 'Confirmed'

group by f.CondoID, b.BookingID)

as b on b.CondoID = c.CondoID

group by c.CondoID;

Cedric queries

/\*This query retrieves the latest feedback for each feedback category.  
Description: This query retrieves the most recent piece of feedback for each feedback category. It does this by first determining the latest feedback ID (MAX(FbkID)) within each category and then joining this information with the Feedback and FeedbkCat tables to get the details of this latest feedback. The result is a list of the latest feedback descriptions along with their corresponding categories, sorted by feedback category.\*/  
SELECT   
    fc.FbkCatDesc,  
    f.FbkDesc,  
    f.FbkDateTime  
FROM   
    Feedback f  
INNER JOIN   
    FeedbkCat fc ON f.FbkCatID = fc.FbkCatID  
WHERE   
    f.FbkID IN (SELECT MAX(FbkID) FROM Feedback GROUP BY FbkCatID)  
ORDER BY   
    fc.FbkCatDesc;

/\*Monthly Facility Booking Counts - This query calculates the total number of bookings for each facility per month. It groups the data by year, month, and facility name, and counts the total number of bookings in these groups. This provides a monthly breakdown of facility usage.\*/  
SELECT   
    YEAR(b.BookingDate) AS Year,  
    MONTH(b.BookingDate) AS Month,  
    f.FacName,  
    COUNT(\*) AS TotalBookings  
FROM   
    Booking b  
INNER JOIN   
    Facility f ON b.FacID = f.FacID  
GROUP BY   
    YEAR(b.BookingDate), MONTH(b.BookingDate), f.FacName  
ORDER BY   
    Year, Month, f.FacName;

/\*Cancelled Bookings and Related Feedback Categories - This query focuses on bookings that have been cancelled. It associates these bookings with feedback categories, aiming to understand the types of feedback related to cancellations. It counts the number of cancelled bookings for each facility and organizes them by feedback category.\*/  
SELECT   
    f.FacName,  
    fc.FbkCatDesc,  
    COUNT(DISTINCT b.BookingID) AS CancelledBookings  
FROM   
    Booking b  
JOIN   
    Facility f ON b.FacID = f.FacID  
JOIN   
    Feedback fb ON b.AccID = fb.ByAccID  
JOIN   
    FeedbkCat fc ON fb.FbkCatID = fc.FbkCatID  
WHERE   
    b.BookingStatus = 'Cancelled'  
GROUP BY   
    f.FacName, fc.FbkCatDesc  
ORDER BY   
    CancelledBookings DESC;

 WeiJun Queries

query1: tget total count of items for sale and average price of the items for each item category using scalar function and join

SELECT IC.ItemCatID, IC.ItemCatDesc,

AVG(IR.ItemPrice) AS AveragePrice

FROM ItemCategory IC

JOIN ItemRelated IR ON IC.ItemCatID = IR.ItemCatID

GROUP BY IC.ItemCatID, IC.ItemCatDesc

ORDER BY AveragePrice DESC;

query 2: get item categories with the highest average item price using subqueries

SELECT IC.ItemCatID, IC.ItemCatDesc,

(SELECT AVG(IR1.ItemPrice)

FROM ItemRelated IR1

WHERE IR1.ItemCatID = IC.ItemCatID) AS AveragePrice

FROM ItemCategory IC

ORDER BY AveragePrice DESC;

query 3: find average number of likes received by each message typeSELECT M.MsgType,

COUNT(L.AccID) AS LikeCount

FROM Message M

LEFT JOIN Likes L ON M.MsgID = L.MessageID

GROUP BY M.MsgType

ORDER BY LikeCount DESC;

Thomas Queries

1. List how many facilities is associated with each condo in descending order of number of facilities.

Select f.CondoID, count(f.FacName) as 'Number of associated facilties' from Facility f

inner join Condo c

on c.CondoID=f.CondoID

group by f.CondoID

order by count(f.facname) desc

1. List the duration of contract for each tenant in detailed in ascending order.

SELECT a.CondoID,AccName ,DATEDIFF(DAY, ContractStartDate, ContractEndDate) AS 'Contract Duration'

FROM Tenant t

inner join account a

on a.AccID=t.TenantID

inner join condo c

on c.CondoID=a.CondoID

order by DATEDIFF(DAY, ContractStartDate, ContractEndDate) asc

1. Find which day , which facility is the more popular choice.

select facname as 'Facility Name',count(facname) as 'Number of Bookings', FORMAT(CONVERT(DATE, SlotDate) , 'dd MMMM yyyy') AS 'Facility Date Slot' from Booking b

inner join Facility f

on b.FacID=f.FacID

group by facname,CONVERT(DATE, SlotDate)

order by count(facname) desc