F28DM Database Management Systems Coursework 1
Database Design and Implementation
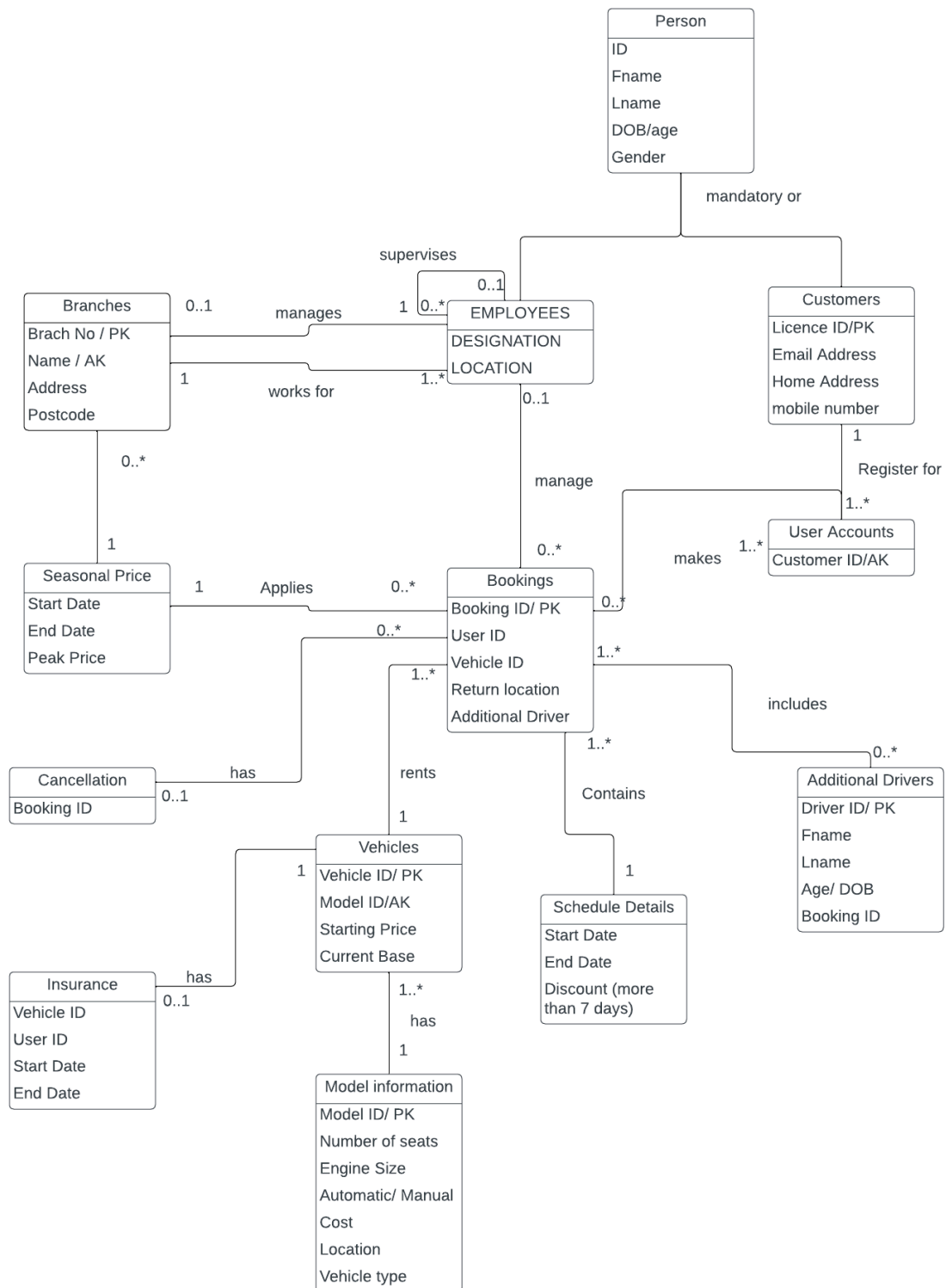
Group 16:

Fahima Patel, Irsa Muhammad Arshad, Josna Akter, Mohammed Rizwan, Tala Hamdan

INDEX

Task 1: Designing and drawing an ER-diagram that captures the requirements.

Person
- ID
- Fname
- Lname
- DOB/age
- Gender

mandatory or

supervises

0..1

Branches
- Brach No / PK
- Name / AK
- Address
- Postcode

0..1   manages

1   0..*

EMPLOYEES
- DESIGNATION
- LOCATION

1

Customers
- Licence ID/PK
- Email Address
- Home Address
- mobile number

1   works for   1..*

0..1

manage

Register for

1

1..*

User Accounts
- Customer ID/AK

0..*

0..*   makes   1..*

0..*

Seasonal Price
- Start Date
- End Date
- Peak Price

1   Applies   0..*

Bookings
- Booking ID/ PK
- User ID
- Vehicle ID
- Return location
- Additional Driver

0..*

1..*

includes

0..*

Cancellation
- Booking ID

has   0..1

rents

1..*

Contains

1..*

1..*

Additional Drivers
- Driver ID/ PK
- Fname
- Lname
- Age/ DOB
- Booking ID

0..*

Vehicles
- Vehicle ID/ PK
- Model ID/AK
- Starting Price
- Current Base

1

1

Schedule Details
- Start Date
- End Date
- Discount (more than 7 days)

1

Insurance
- Vehicle ID
- User ID
- Start Date
- End Date

has   0..1

1..*   has   1

Model information
- Model ID/ PK
- Number of seats
- Engine Size
- Automatic/ Manual
- Cost
- Location
- Vehicle type

<u>Requirements not mentioned on the erd:</u>

- Vehicles can be rented from a base and returned to another rental location (i.e 1 way rentals).

- Each vehicle has a home base where it will be serviced. All service schedules for the vehicles will be handled by this system so that they know when cars are not available, but car parts, mechanics time, and detailed service reports will be handled by a separate system.

- They only rent to the public, not commercial rentals.

- The minimum rental period is 1 day and rentals are whole numbers of days.

- All cars can be driven on a standard UK driving licence.

- The company policy sets a minimum age to rent a car as 25 years, and 30 years to hire an MPV, van, sports or luxury vehicle.

- Rental prices will vary during the seasons (i.e. peak time prices).

- Any booking which is cancelled will be removed from this system, however this will not impact user registrations for the web/mobile app booking system.

- The website/app will show a picture of the actual car being hired with each booking.

- Assume that all cars can be moved to any UK location overnight based on a car-carrier service.

- The person that makes the booking must be the main driver

- All bookings will be made via web page/mobile app which requires the user to register.

- During servicing periods, vehicles are unavailable for rental.

Task 2: Translating the ER-diagram into a relational schema.

Strong entities:

Bookings (bookingID, pID, vehicleID, returnLocation, additionalDriver, seasonalPrices, scheduleDetails)
Branches (brachID, name, address, postcode, seasonalPrices)
Vehicle (vehicleID, modelID, startingPrice, currentBase)
Model info (modelID, seatNumber, engineSize, automatic/manual, cost, location, vehicleType)

Specialisation / generalisation, mandatory or:

Employee (pID, fName, lName, DOB, gender, designation, branchID)
Customer (pID), fName, lName, DOB, gender, licenceID, email, homeAddress, mobileNumber)

Weak entities:

User account (pID, mainDriverID, bookingID)
Seasonal prices (seasonalPrices, startDate, endDate, peakPrice)
cancellation (bookingID)
Insurance (vehicleID, pID, startDate, endDate)
Schedule details (bookingID, startDate, endDate, discount)
Additional drivers (pID, fName, lName, DOB, bookingID)

M:N relationships:
booking_additionalDrivers(bookingID , pID) [bookings + additional drivers]
user_bookings(pID, bookingID) [user accounts + bookings]

Data Dictionaries:

Bookings

Attribute	Description	Domain	null?	Primary key	Foreign key
bookingID	Booking ID	int(6)	N	Y	
pID	Person ID	int(6)	N	N	Employee.pID
vehicleID	Vehicle ID	varchar(20)	N	N	Vehicle.vehicleID
returnLocation	Company site where is returned	varchar (20)	N	N	
additionalDriver	Name of additional driver	text(20)	Y	N	
seasonalPrices	Seasonal prices	decimal(8,2)	Y	N	
scheduleDetails	Dates booked	varchar(255)	N	N	

Branches

Attribute	Description	Domain	null?	Primary key	Foreign key
brachID	Branch ID	int(6)	N	Y	
name	Name of the branch	varchar(30)	N	N	
address	Branch address	varchar (30)	N	N	
postcode	Postal code	varchar (10)	N	N	
seasonalPrices	Seasonal prices	decimal(8,2)	Y	N	SeasnolPrices. seasonalPrices

Vehicle

Attribute	Description	Domain	null?	Primary key	Foreign key
vehicleID	Vehicle ID	varchar (20)	N	Y	
modelID	Model ID	varchar (20)	N	N	ModelID.model ID
startingPrice	Starting price of vehicle rent	decimal(8,2)	N	N	

currentBase	Base where vehicle is currently located	varchar (20)	N	N	

Model info					
Attribute	Description	Domain	null?	Primary key	Foreign key
modelID	Model ID	varchar (20)	N	Y	
seatNumber	Number of seats	Int(1…15)	N	N	
engineSize	Engine size	varchar (10)	N	N	
automatic/manual	Whether the car has automatic or manual controls	text(10)	N	N	
cost	Price of the model	decimal(8,2)	N	N	
location	Site where the model is available	varchar (20)	N	N	
vehicleType	Type of vehicle	varchar(20)	N	N	

Employee					
Attribute	Description	Domain	null?	Primary key	Foreign key
pID	Person ID	int(6)	N	Y	
fName	First name	text(20)	N	N	
lName	Last name	text(20)	N	N	
DOB	Date of birth	Date	N	N	
gender	Employee's gender	enum[M|F]	N	N	
designation	The job of the employee	text(20)	N	N	
branchID	Branch ID	int(6)	N	N	BrachID.branchID

Customer

Attribute	Description	Domain	null?	Primary key	Foreign key
pID	Person ID	int(6)	N	Y	
fName	First name	text(20)	N	N	
lName	Last name	text(20)	N	N	
DOB	Date of birth	date	N	N	
gender	Customer gender	enum[M|F]	Y	N	
licenceID	Licence ID	varchar (20)	N	N	
email	Email address	varchar (30)	N	N	
homeAddress	home address	varchar (50)	N	N	
mobileNumber	phone number	varchar(15)	N	N	

User account

Attribute	Description	Domain	null?	Primary key	Foreign key
pID	Person ID	int(6)	N	Y	
bookingID	Booking ID	int(6)	N	N	BookingID.bookingID

Seasonal prices

Attribute	Description	Domain	null?	Primary key	Foreign key
seasonalPrices	The seasonal prices	Decimal(8,2)	N	Y	
startDate	Start date of seasonal discount	date	N	N	
endDate	End date of seasonal discount	date	N	N	

peakPrice	Peak price	decimal(8,2)	Y	N	

cancellation					
Attribute	Description	Domain	null?	Primary key	Foreign key
bookingID	Booking ID	int(6)	N	Y	BookingID.booking ID

Insurance					
Attribute	Description	Domain	null?	Primary key	Foreign key
vehicleID	Vehicle ID	varchar (20)	N	Y	vehicle.vehicle ID
pID	Person ID	int(6)	N	N	Customer.pID
startDate	Start date	date	N	N	
startDate	End date	date	N	N	

Schedule details					
Attribute	Description	Domain	null?	Primary key	Foreign key
bookingID	Booking ID	int(6)	N	Y	BookingID.booking ID
startDate	Start date of rental	date	N	N	
endDate	End date of rental	date	N	N	
discount	discount	decimal(8,2)	N	N	

Additional drivers					
Attribute	Description	Domain	null?	Primary key	Foreign key
pID	Person ID	int(6)	N	Y	
fName	First name	text(20)	N	N	

lName	Last name	text(20)	N	N	
DOB	Date of birth	date	N	N	
bookingID	Booking ID	int(6)	N	N	BookingID.boo
kingID

Task 3: Implementation of the schema in MariaDB.

```sql
-- Create the database
CREATE DATABASE IF NOT EXISTS car_rental_db;
USE car_rental_db;


SET storage_engine=InnoDB;

-- Create Branches table
CREATE TABLE Branches (
    branchID INT(6) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(30) NOT NULL,
    address VARCHAR(30) NOT NULL,
    postcode VARCHAR(10),
    seasonalPrices DECIMAL(8,2),
    CONSTRAINT fk_seasonalPrices FOREIGN KEY (seasonalPrices) REFERENCES
SeasonalPrices(seasonalPrices) ON DELETE SET NULL
) ENGINE=InnoDB;

-- Create ModelInfo table
CREATE TABLE ModelInfo (
    modelID VARCHAR(20) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    seatNumber INT CHECK (seatNumber >= 1 AND seatNumber <= 15),
    engineSize VARCHAR(10),
    transmission ENUM('automatic', 'manual'),
    cost DECIMAL(8,2),
    location VARCHAR(20),
    vehicleType VARCHAR(20)
) ENGINE=InnoDB;

-- Create Vehicle table
CREATE TABLE Vehicle (
    vehicleID VARCHAR(20) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    modelID VARCHAR(20),
    startingPrice DECIMAL(8,2) NOT NULL,
    currentBase VARCHAR(20) NOT NULL,
    CONSTRAINT fk_modelID FOREIGN KEY (modelID) REFERENCES ModelInfo(modelID)
ON DELETE SET NULL
) ENGINE=InnoDB;

-- Create Employee table
CREATE TABLE Employee (
    pID INT(6) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    fName TEXT(20) NOT NULL,
    lName TEXT(20) NOT NULL,
    DOB DATE NOT NULL,
```

```sql
    gender ENUM('M', 'F') NOT NULL,
    designation VARCHAR(20) NOT NULL,
    branchID INT(6),
    CONSTRAINT fk_branchID FOREIGN KEY (branchID) REFERENCES
Branches(branchID) ON DELETE SET NULL
) ENGINE=InnoDB;

-- Create Customer table
CREATE TABLE Customer (
    pID INT(6) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    fName VARCHAR(20) NOT NULL,
    lName VARCHAR(20) NOT NULL,
    DOB DATE NOT NULL,
    gender ENUM('M', 'F') NOT NULL,
    licenceID VARCHAR(20) NOT NULL,
    email VARCHAR(30),
    homeAddress VARCHAR(50),
    mobileNumber INT(15)
) ENGINE=InnoDB;

-- Create Bookings table
CREATE TABLE Bookings (
    bookingID INT(6) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    pID INT(6),
    vehicleID VARCHAR(20),
    returnLocation VARCHAR(20),
    additionalDriver VARCHAR(20),
    seasonalPrices DECIMAL(8,2),
    scheduleDetails VARCHAR(255),
    FOREIGN KEY (pID) REFERENCES Employee(pID) ON DELETE SET NULL,
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID) ON DELETE SET NULL
) ENGINE=InnoDB;

-- Create Insurance table
CREATE TABLE Insurance (
    vehicleID VARCHAR(20),
    pID INT(6),
    startDate DATE,
    endDate DATE,
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID) ON DELETE CASCADE,
    FOREIGN KEY (pID) REFERENCES Customer(pID) ON DELETE CASCADE,
    PRIMARY KEY (vehicleID, pID)
) ENGINE=InnoDB;

-- Create AdditionalDrivers table
CREATE TABLE AdditionalDrivers (
    pID INT(6),
    fName VARCHAR(20) NOT NULL,
```

```sql
    lName VARCHAR(20) NOT NULL,
    DOB DATE NOT NULL,
    bookingID INT(6),
    FOREIGN KEY (bookingID) REFERENCES Bookings(bookingID) ON DELETE
CASCADE,
    PRIMARY KEY (pID, bookingID)
) ENGINE=InnoDB;



-- Create table for User_account
CREATE TABLE User_account (
    pID INT(6) NOT NULL AUTO_INCREMENT,
    bookingID INT(6) NOT NULL,
    PRIMARY KEY (pID),
    FOREIGN KEY (bookingID) REFERENCES Schedule_details(bookingID) ON DELETE
CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;




-- Create SeasonalPrices table
CREATE TABLE SeasonalPrices (
    seasonalPrices INT(6) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    startDate DATE,
    endDate DATE,
    peakPrice DECIMAL(8,2)
) ENGINE=InnoDB;

-- Create Cancellation table
CREATE TABLE Cancellation (
    bookingID INT(6) NOT NULL PRIMARY KEY,
    FOREIGN KEY (bookingID) REFERENCES Bookings(bookingID) ON DELETE
CASCADE
) ENGINE=InnoDB;

-- Create ScheduleDetails table
CREATE TABLE ScheduleDetails (
    bookingID INT(6) NOT NULL PRIMARY KEY,
    startDate DATE,
    endDate DATE,
    discount VARCHAR(4),
    FOREIGN KEY (bookingID) REFERENCES Bookings(bookingID) ON DELETE
CASCADE
) ENGINE=InnoDB;
```

INDEXES
1. *CREATE INDEX start_end_date ON ScheduleDetails(startDate, endDate);*
   ● Explanation: Starting date and the ending date of the ScheduleDetails table are used as a composite index to help filter all the bookings within the specified time range.

2. *CREATE INDEX vehicle_type ON ModelInfo(vehicleType);*
   ● Explanation :VehicleType of ModelInfo table is used as an index which filters out all the vehicle models based on their types.

3. *CREATE INDEX designation_employee ON Employee(designation);*
   ● Explanation : Designation of the Employee table is used as an index where all the employees can be sorted based on their job titles.

4. *CREATE INDEX model_cost ON ModelInfo(cost DESC);*
   ● Explanation : Cost of ModelInfo table is taken as an index which involves sorting vehicle models based on their cost in descending order making it useful for budget related constraints .

5. *CREATE INDEX insurance_dates ON Insurance(startDate, endDate);*
   ● Explanation : The start Date and end Date of Insurance table are used as a composite index that involves filtering insurance policies or the active insurance policies within a time range.