# Group Members:

|  |  |  |
| --- | --- | --- |
| **Name** | **Student Number** | **Programme Code** |
| Erifeoluwa O. Jamgbadi | C18387973 | DT282/2 |
| Djibril Coulybaly | C18423664 | DT228/2 |
| Allyanna Riyann Cruz | C18310641 | DT282/2 |

# 

# Domain Types:

|  |  |  |  |
| --- | --- | --- | --- |
| **Domain Type Name** | **Type Definition** | **Reasons Introduced** | **Examples** |
| Address | VARCHAR(320) | For attributes which are character-based addresses | location\_addr in Car |
| Amount | NUMERIC(7,2) | For attributes which represent monetary amounts | x\_amt in Final\_Bill |
| Description | VARCHAR(50) | For attributes which are character-based descriptions | type\_desc in Contract |
| ID | NUMERIC(50) | For attributes which represent a unique numeric identifier | num\_id in Branch |
| mileage | NUMERIC(7,2) | For attributes which represent the mileage that the car has driven | car\_mil in Car |
| Name | VARCHAR(30) | For attributes which are character based full names | person\_name in Customer |
| Time | TIMESTAMP WITH LOCAL ZONE | Date and time attributes which require a very clearly defined time element | date\_occur in Contract |

# Major Decisions: Decision 1: Sale and Offline Sales

We had to discuss what attributes, we were going to put into online sales and offline sales. We ended up deciding to change offline sales to just sale because sales made offline are sales made in the branch and in real life, those will just be considered as sales.

With regards to offline sales(which is just sale now),  we had to discuss what attributes that was going to be included as part of sale and online sales. This was an important question, we had to discuss among ourselves because some of the important points, we raised was that most of the attributes in online sale were going to be foreign keys because online sales is connected to many attributes in other tables, such as the Customer(we also wanted to record the customer id, since each customer will have their own unique id, so it will be easier to know which order went with each customer), Employee(we have to record the employee, that worked with the transaction) tables.

Another decision, we had to make was whether we were going to delete the offline sales and just have both offline and online sales in one table called sales. In the end, we decided to just leave it as two separate tables.

Lastly, the last decision, we discussed for the online sale and sale table was, which one we were going to connect to contract. We decided to connect the sale table to the contract and customer table because online sale has a relation with sale, made available through the attribute sale id, so with the sale id connecting both table, since it is the sale id, it would be able to allow us to access the contract for the sale and customer of the sale.

**Decision 2: Relationships - Cardinality**   
  
We chose the cardinality between:  
  
Final\_bill and Contract as one to one because there can one contract and only one final bill. (i.e a customer can choose to rent cars on different periods of time and choose a different car but the contract will be a different contract and each renting period will have a final bill).

Customer and Contract as one to many because a customer can have more than one contract (i.e a Customer can have multiple renting periods and choose different cars).

Car and Contract as one to many because one car can have many contracts (i.e a car can be rented by different people).

Sale and Contract as one to one because each sale means a new contract must be made. Each sale requires a new contract, this specifies the rental period and car to be rented - even if it’s the same customer who wants to rent a car in two different rental periods, a new contract is needed.

Branch and Contract as one to many because in one branch many contracts can be made.

Branch and Employee as one to many because one branch can have many employees.

Employee and Sale as one to many because an employee can have many sales.

Sale and Online\_sales  as one to one because an online sale that is made means a sale is made. Online\_sales needs sale\_id from sale. We also decide to join the Customer table and Sales as the Customer table has the relevant information for Sales.

# Decision 3: Entities and Tables

We chose the 8 entities ( Final\_bill, Contract, Sale, Online\_sale, Car, Customer, Branch and Employee) because we read through the assignment case study and we created the entities based on each of the paragraphs. We thought that the following entities we chose contained the attributes that was required and grouped them appropriately.

# Decision 4: Domain Type

We have 7 domain types, which are Mileage, Description, Name, ID ,Amount Address, Time.

*Reasons why we decided to have each domain type:*

Mileage:  we decided to have mileage because mileage needed to be documented in table such as Contract and Car.

Description: We decided to have a description domain type because we needed a domain type, that could be used by any attributes, that wanted to describe anything to do with just characters.

Name: We had name as a domain type because we wanted the customer’s name to have its own distinguished domain type because we thought, it was different from the description domain type.

ID: We needed an id domain type because id was used by many tables such as sale, Contract, branch and we wanted it to have its own distinguished numeric domain type.

Amount: We needed to have a domain that recorded money or attributes that got to do with money/cost. This domain type was used by the attributes in Contract and Final

Address: We decided to have an address domain type because the size and logical type is different from Description, because in an address number and characters are used and the description domain type was only limited to characters. The length of the address was also more than the description.

Time: we need to have time as a domain type because we need a domain type to be able to record the time and date and this was needed by some of the attributes in Contract and Employee.

# Decision 5: Primary and Foreign Keys

In order for our database and tables to be fully functional and ensuring that each record in a table is uniquely identified, we needed to assign primary and foreign keys to each of the tables.

Primary keys are used to identify a table record. There can only be one primary key per table A primary key must have data in the table row that’s unique and that table data cannot be left blank or NULL.

Foreign keys are used to refer to a primary key in another table, acting as a cross reference between tables. Unlike primary keys, multiple foreign keys can be introduced in a table

As a group, we decided on the following primary and foreign keys for our tables:

|  |  |  |
| --- | --- | --- |
| **Table** | **Primary Key + Reason** | **Foreign Key + Reason** |
| Final\_bill | 1. **amount\_owed:** Each customer will owe the company a different amount of money depending on their contract. | No foreign keys used in this table |
| Contract | 1. **contract\_id:** Each customer will need to have a unique contract with the company for every car they hire | 1. **Customer\_customer\_id:** Need to identify which customer this contract belongs to from the Customer table 2. **Car\_registration\_number:** Need to identify which car was used in relation to the customer from the Car table 3. **Branch\_branch\_id:** Need to identify which branch the car was rented from using the Branch table 4. **Sale\_sale\_id:** Need to identify which sale was made using the Sale table 5. **Final\_bill\_amount\_owed:** Used to identify how much the customer owes the company  after usefrom the Final\_bill table |
| Branch | 1. **branch\_id:** The company has multiple branches, so a primary key is needed to distinguish which branch is which | No foreign keys used in this table |
| Employee | 1. **employee\_id:** Within the company, there are many employees hired. To distinguish who is who, an Id must be created for each employee | 1. **Branch\_branch\_id:** Need to identify which branch the car was rented from using the Branch table |
| Online\_sales | 1. **transaction\_state:** | 1. **Sale\_sale\_id:** Need to identify which sale was made using the Sale table |
| Sale | 1. **sale\_id:** Every sale of a car for hire will be different from customer to customer. To avoid confusion on which sale belongs to which customer, and to distinguish which employee carried out the sale, a primary key is used to identify which sale belongs to which employee | 1. **Employee\_employee\_id:** Need to identify which employee carried out the sale using the Sale table |
| Car | 1. **Registration\_number:** Every car has a different registration number, which can give the necessary details about the car such as the make, model, fuel type etc. | No foreign keys used in this table |
| Customer | 1. **customer\_id:** There may be customers who have the same name or personal details as another customer. This primary key will be used to identify which customer is which | No foreign keys used in this table |

# Decision 6: Constraints

 Finally, our last major decision was to decide on the constraints to implement. Constraints are rules that are enforced to the data in the table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database. In our tables, our primary and foreign keys are constraints that we've implemented already. However we used CHECK constraints to check if the customer email ended in @\_\_\_\_. Com **(customer\_email like '%@%.com')**. We’ve also added a constraint check that the customer was over 18 to buy a car, so we used this check constraint **(customer\_age >= 18).**

Our final constraint was **(employee\_email\_address like '@speedyrentals.com'),** which we used to check if the employee’s used their company email.