CODE BASE

PROFESSIONAL DATABASE DESIGNERS

CMPG311

CODE BASE

PHASE 1,2 &3

INITIAL STUDY, DATABASE DESIGN & PHYSICAL DESIGN

Contents

Project Phase 1- Initial Study

1. Introduction

1.1. Project Members

Member 1

Initials and Surname: NL MOSIA

Student number: 29905184

Role: Leader

Member 2

Initials and Surname: T MUNHENGA

Student number: 31671853

Role: N/A

Member 3

Initials and Surname: LMG NENE

Student number: 32642156

Role: N/A

Member 4

Initials and Surname: SR MAZIBUKO

Student number: 31060552

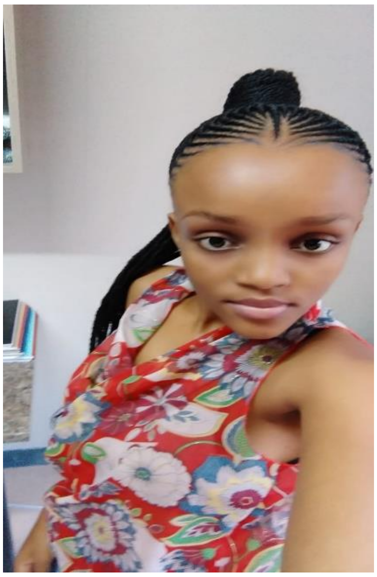
Role: N/A

Member 5

Initials and Surname: NC HLATSHWAYO

Student number: 32571259

Role: N/A



Member 6

Initials and Surname: NE NZANZA

Student number: 32210310

Role: N/A

Member 7

Initials and Surname: TG MOTSOENENG

Student number: 30208394

Role: N/A

1.2. Background Information

Gallant Automobiles is an Auto dealership company that is located in Vanderbjilpark, South Africa. The company has been operating for 4 years, it was officially opened in 2017. The company serves to offer proficient benefits and encounters in car purchases to clients.

Gallant Automobiles has a prevalent use of primitive data storage and management methods such as paper filing systems and the resulting core of storage is MS Excel spreadsheets and this makes the company to have tons of spreadsheets which is very inconvenient and leads to islands of data and redundancy. The management has time management problems because they have to search hundreds of spreadsheets to get the correct data for report generation.

Gallant Automobiles imports cars from various countries. In this regard, there is data inaccuracy and management in outputting the correct details of the transaction such as arrival time of the vehicle and the provision of unexpected faulty cars to clients.

Gallant Automobiles has a non-timely way of checking car availability. This is done manually and often result in human error.

2. Company Situation Analysis

2.1. Company Objectives

- Increasing profit levels.
- Establishing healthy business relationships with customers.
- Engage new potential customers.
- Providing good services and products(cars).
- Expanding and developing the company into a franchise.
- Increasing efficiency, effectiveness and productivity.
- Reduce error rates.

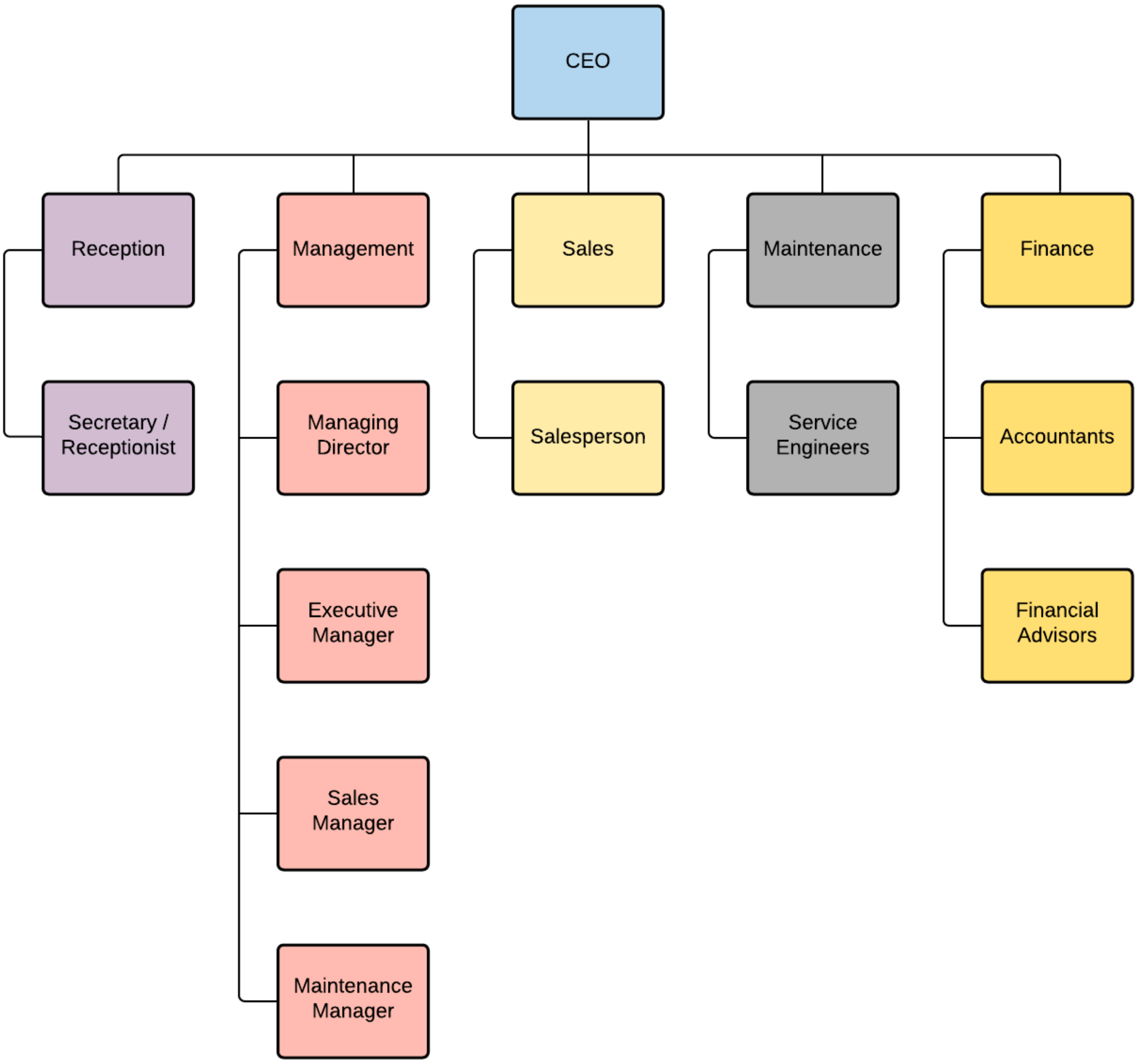
2.2. Company operations

This is how Gallant Automobiles conducts business operations on a daily basis:

Business usually starts at 06:00 and ends at 18:00 from Monday to Friday. Every morning the staff members head to work to do each of their duties. The receptionist answers phone calls from existing or potential clients. A client may call to ask if a certain car is available. The receptionist then goes to the car sales person who has the necessary information. This person is responsible for giving the client all the necessary information regarding the car and its sale. The sales person then physically checks at the garage if the desired car is there or not. If the car is unavailable then the company tries to get the required car by trade-offs, buying from people or car-manufacturing companies. If the car is available, then he may ask the client which payment option he prefers. The sales person then contacts the finance department. This department takes the clients' personal information and banking details and sends it to the bank for approval of the client's application. Once the application is approved, the client called to the dealership and is given binding forms of which he should sign. The client then receives their car and may drive off. The customer may also request a test drive from the car sales person before purchasing it or signing any forms. Once all transactions are processed, the car sales person compiles a report and also stores the details and information on the company's conventional file system. The report is then sent to management which will analyse these reports and to necessary activities.

2.3. Company Organisational Structure

2.3.1. An Overview of the Organisational Structure



1. Reception

Reception is important because receptionists are basically the confront and the voice of the dealership, and for that reason they play a greatly critical part within the representation of the dealership because they welcome the guests and customers. Receptionists answer and forward phone calls, they maintain calendars and take messages, they increment the security by deciding who gets to the offices of employees in the dealership, they complete necessary paperwork and file any paperwork of the customers and employees.

2. Management

Management is essential for the dealership so that the dealership will be able to achieve its goals and objectives through performing managerial tasks done by a manager employed in a dealership. A manger in key position should perform capacities like arranging (planning) the operations that have to be followed in the dealership, organizing each and every operation taking place in the dealership, staffing all the necessary assets and resources that the dealership uses and needs to operate, setting goals for the dealership to operate smoothly, motivating all the employees to work at their optimum to ensure the success of the dealership, coordinating and controlling different levels of the dealership to ensure that the dealership becomes sustainable enough.

3. **Sales**

   The sales department plays an urgent part within the dealership by ensuring the success of the dealership whereby the salesperson is mindful and responsible for welcoming the customers, helping customers discover the products/services that the dealership provides and ensuring products/services purchases. The role of a salesperson is to lead conversations to ensure that customers are happy with the product/service and to ensure customer retention by ensuring the purchase of the product/service which will result in dealership growth and surplus of profits.

4. **Maintenance**

   Maintenance department in the dealership is a critical component of a well-functioning production of products. It assists the dealership to keep up their resources as well as the resources of the customers while controlling the time and costs to guarantee the greatest proficiency of fabricated products, utilities and other related facilities. This department makes a difference to ensure and secure a dependable and palatable quality of the production, because poorly maintained facilities can cause flimsiness and mostly it could lead to customers being unsatisfied therefore maintenance and service engineers will ensure that the matter stays on track by maintaining and servicing each and every vehicle available in the dealership.

5. **Finance**

   Finance department is one of the exceptionally inconspicuous segments of a dealership that can make or break the vision and goals of the dealership. All companies require finance department for everyday operations to continue. A finance department in the dealership will ensure profit creation to sustain the dealership for the future and for the next generation. It will ensure the operational expenses where all the expenses incurred for the dealership to operate are being covered and paid for. It will ensure asset creation in the dealership by purchasing all the required assets using some of the profits made by the dealership. It will ensure cash flow management and whether all of the employees in the dealership are paid their salaries.

3. Problems and Constraints

3.1. Problems

- **Bad management:** Poor communication between workers in managerial positions. Our suppliers may sometimes take time to deliver cars ordered by our company so that we can quickly give them to our customers on the agreed date and time. We may forget to inform our customers of the delay, resulting in bad communication and how we do things as a whole.

- **Provision of faulty cars:** Gallant Automobiles may possibly provide customers with faulty cars, which had not been previously checked from our suppliers. This will cause the customers not to trust us with the services we render.

- L**osing the customers' loyalty**: Inconsistent deliveries of cars from suppliers may result in losing the customers' loyalty. The customers will think of somewhere else to go for their purchases instead of coming to our store. Less customers will result in less profit.

- **Writing reports on paper**: Using the manual filing system will only delay the process of our customers getting what they purchased on time. Automatic filing system need to be introduced to rush the process and make the customers remain purchasing from our store. Everything should be computer based.

- **Struggle to keep track of unavailable cars**: The struggle to keep track of unavailable cars, because the auto dealership is still using a manual filling system. If the automatic filing system is introduced, it is going to be easier to check for available cars. Employees will just type in the keyword of the wanted car in the 'search' textbox.

- **Data redundancy**: New files are created constantly due to the disorganised structure for file storage, which leads to loss or misplacement of files that contain confidential information of customers and resulting in the same data entered twice. By this a customer will question the reliability and security of their personal data in the hands of the company. Customers will not continue purchasing at a store where their data get lost out of the blue.

- **Scheduling and ordering is done via phone calls then recorded on basic platforms, such as Microsoft Word**: Ordering via phone calls is inconvenient, because the staff/employees will think they heard the customers correctly, and jot down what is not wanted.

- **There is no record for stolen cars:** Stolen cars are not recorded. This could lead in loss of revenue and the inventory (cars) security being compromised.

3.2. Constraints

- A company has a certain number of cars it can sell.
- Only 100 orders can be made in a day.
- Customers pay R150 extra if they took time to collect their cars after it was announced that they were available.
- Each car type has 500 cars.
- Each car type is managed by 5 employees.

4. Database System Specifications

4.1. Objectives to Solve Identified Problems

- The database must be inclusive of a swift intelligent data retrieval system that will guarantee that management, staff and people concerned with the database are able to complete their work duties on time, resulting in the assurance that suppliers deliver the cars ordered on time and the company also delivers requested cars on the negotiated date and time.
- The database must monitor and capture the amount of faulty cars along with their specific car profile at an instant when cars are being inspected and checked once they arrive from the car suppliers. The database should not allow a car be added to the database if it is not yet inspected.
- The various car dealership departments must know their competitors; how their databases are organized/offer, their strengths and weaknesses. In knowing their competitors, they should also know their customers, step up on the company's marketing strategies, and update the company's image by expanding their offers, database intelligence so that data can be utilized for interpolation and extrapolation of customer trends and preferences all in one.
- The diversified department's records should be stored in a suitable centralized database; this is where each department will be able to comfortably retrieve information. The database must be expendable for the generation of reports which will assist in decision making.
- The car dealership company must abolish the current manual filing system, and replace it with a new database system with a repository that will be automatically reliable and convenient to keep track unavailable cars. This

must be done in the 3rd normalization form to keep away from data redundancy and data anomalies. Thereafter, searching will be made easier.
- The database should have authentication systems to control the accessibility of the records, and the extend that each staff member is permitted to proceed.
- The newly created database must be linked with a newly developed car dealership application/online website that customers can use to place orders of various cars available on site. The associated will make it easier to keep track of orders, availability, purchases and so on.
- The database should consist of a "**STOLEN CARS**" record.

4.2. Information The Company Requires from the Database:

- Customer preferences and trends regarding new car models.
- Which type of cars are available and which are not available.
- Knowing when and how cars will be supplied.
- Knowing the historic customer data.
- The car dealership needs to know the vehicle details and specifications data.
- The company must also know the cost of the car ownership data (e.g. insurance, depreciation, financing, maintenance, state fees).
- The company needs to know when customers have a problem with their newly purchased vehicle(s), so that the issue can be solved as soon as possible.
- Whether or not a customer paid for a car.
- Managers responsible for each type of car.
- Type of car purchased.

4.3. Scope

The scope will assist the designer to improve the database according to the operational requirements that are operating in the company. The database environment will be as follows:

Resources
The database will consist of a department of customers and employees. Customers are our main concern to be able to be helped quickly. The cars that customers purchase will be linked to a brand and the price of the car.

Maintenance
The database will have to be maintenance daily since we have new customers daily and the employees will have different shifts. Also, this will help to keep record if there was a mistake made by the employee then the management will know exactly who was on duty.

Customer Service
The department will have a date where the customer purchases the car. Also, if they have ordered their car online there will be a date of the place of order then when the customer will they will be able to collect their car.

Marketing
There will be a department where customers will be able to rate our customer service skills. The customers can also point out things they will like to be featured. This will be done in different platforms such as social media, company's website, SMS questionnaire.

4.4 Boundaries

- **Budget boundaries:** The database system design will require a budget of R250 000 to fully designed and implemented.

- **Software boundaries:** An operating system specifically Windows 7 or Windows 10 any edition with MS-Office, will be required to support the development of the new database system.

- **Hardware boundaries:** The system will be designed around the current existing hardware. The company used the manual filing system thus making the transition into the new system easy, and saving the designer a great amount of time and effort to implement the new system within the company.

- **Personal boundaries:** There are seven employees responsible for the database development project. They are going to work 30 hours per week excluding weekends in order to completely design and implement the database. The project will take a period of 6 months to be completed.

5. Conclusion

With the improvements that have been mentioned above they will influence the productivity of Auto deal. The implementation of a database will help with time management as customers will be helped quickly. The level of customer's service earns the Auto deal a revenue. Also, there will be less data redundancy, for there will no duplicate on the database for there will be unique and act as a counter(customers) we are able to help per day. The good thing is about the new system is that it will create a strong infrastructure for the company to be able to store data information and data more sufficiently.

Project Phase 2 – Database Design

Conceptual Design

Business Rules for Gallant Automobiles

Day to day operations

The day to day operation of Gallant Automobiles can be divided into six (6) segments and are as follows:

1. Client request
2. Client data analysis
3. Vehicle selection process
4. Financial analysis
5. Binding agreement
6. Report generation

Business rules on Client request

- A client may be attended by a single receptionist.
- A receptionist may only attend to one client a time.
- A receptionist records the details of the request.
- The request is stored by the receptionist in relevant records

Business rules on Client Data Analysis

- A receptionist may request information from one or more sales people
- Each sales person can receive a request from only one receptionist.
- The sales person generates one or more reports
- The report is generated by a sales person.

Business rules on Vehicle Selection process

- Each sales person may check the availability of one or more cars.
- A car's availability may be checked by one or more sales person.
- A sales person can generate a report on the condition of one or more cars.
- A car's condition report may be generated by a sales person.

Business rules on Financial Analysis

- A sales person can contact the finance department.
- The finance department can be contacted by a one or more sales people.
- The finance department requests information from the bank.
- The bank may receive a request from a single department.

Business rules on Binding Agreement

- A client can sign only one form.

- A form can be signed be signed by one or more clients.

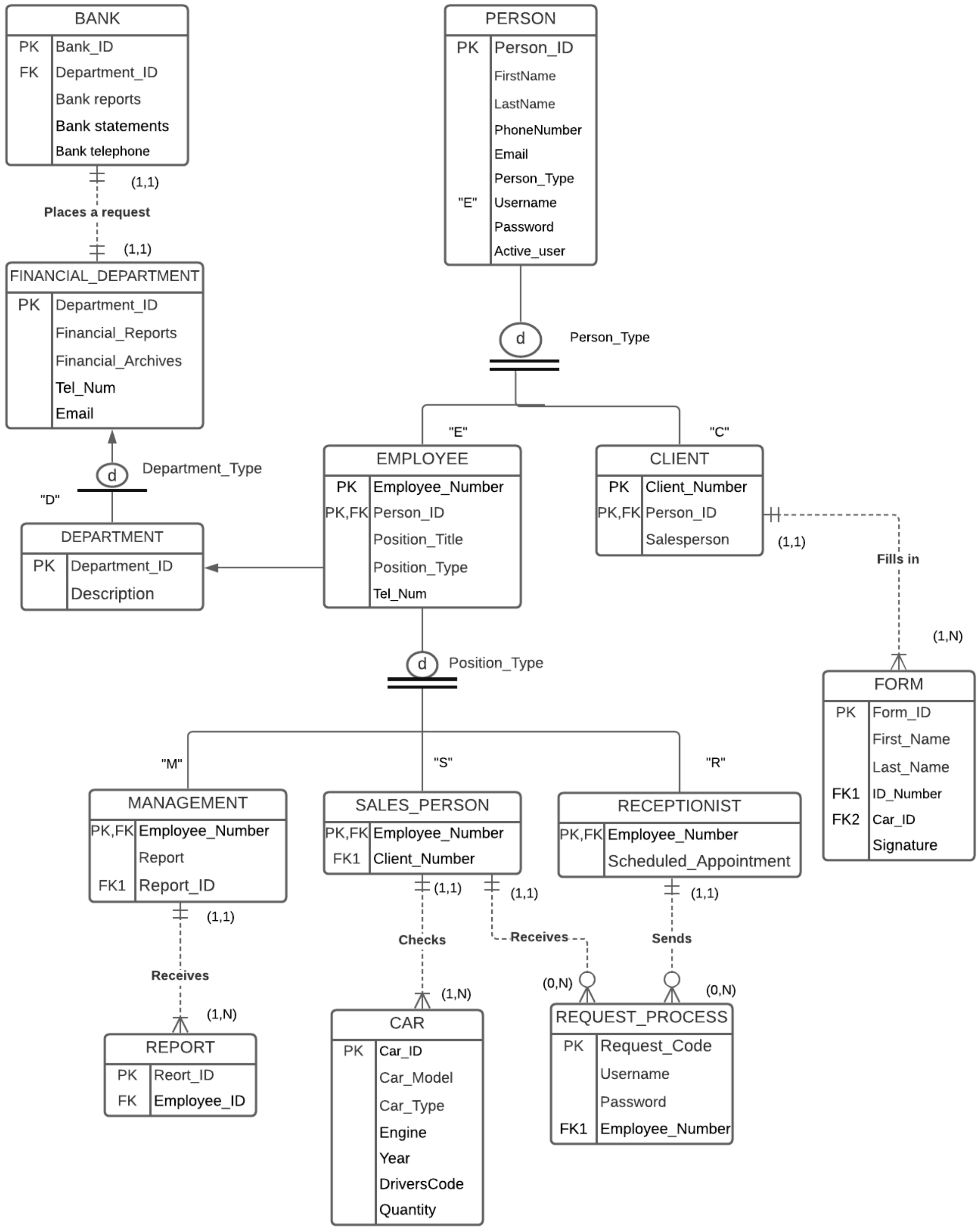
Business rules on Report Generation

- A sales person can compile one or more reports.
- A report can be compiled by only one salesperson.
- A report can be sent to the management
- Management can receive one or more reports.

Final Business rules for Gallant Automobiles

- A client may be attended by a single receptionist.
- A receptionist may only attend to one client a time.
- A receptionist may request information from one or more sales people
- Each sales person can receive a request from only one receptionist.
- Each sales person may check the availability of one or more cars.
- A car's availability may be checked by one or more sales person.
- A sales person can contact the finance department.
- The finance department can be contacted by a one or more sales people.
- The finance department requests information from the bank.
- The bank may receive a request from a single department.
- A client can sign only one form.
- A form can be signed be signed by one or more clients.
- A sales person can compile one or more reports.
- A report can be compiled by only one salesperson.
- A report can be sent to the management
- Management can receive one or more reports.

Entity Relationship Diagram



Notes on Entity Relationship Diagram

ERD Interpretation

Weak relationships

- Financial_Department to Bank

- Client and Form
- Receptionist and Request_Process
- Salesperson and Request_Process
- Salesperson and Car
- Management and Report
- Employee and Department

Strong relationships

- Person and all of its subtypes (Employee and Client)
- Employee and all of its subtypes (Management, Salesperson and Receptionist)

Optional relationships

- None

Weak entities

- Car
- Receptionist

Strong Entities

- Person
- Bank
- Financial_Department
- Employee
- Client
- Management
- Salesperson
- Department
- Report
- Request_Process

Super type entities

- Person
- Employee

Subtype entities

- Employee
- Client
- Management
- Salesperson
- Receptionist

Composite/bridge entities

- REQUEST_PROCESS

Composite keys

- Employee_Number
- Person_ID

Logical Design:

Map the conceptual model to the logical model

Strong Entities:

- Bank - (Bank_ID[PK], Department_ID[FK], Bank_reports, Bank_statements, Bank_telephone).
- Person - (Person_ID[PK], First_Name, Last_Name, Phone_Number, Email, Gender, Username, Username, Password, Active_User).
- Financial department – (Department_ID[PK], Financial_Reports, Financial_Archives, Telephone_Number, Email).
- Department – (Department_ID[PK], Description).
- Employee – (Employee_Number[PK], Person_ID [PK, FK], Position_Title, Position_Type, Telephone_Number).
- Client – (Client_Number[PK], Person_ID [PK, FK], Salesperson)
- Form – (Form_ID[PK], First_Name, Last_Name, ID_Number[FK1], Car_ID[FK2], Signature).
- Report – (Report_ID[PK], Employee_ID[FK]).
- Car – (Car_ID[PK], Car_Model, Car_Type, Engine, Year, Drivers_Code, Quantity).
- Request process – (Request_Code[PK], Username, Password, Employee_Number[FK]).

Supertype/Subtype Relationships:

- Supertype: Person and Subtypes: Employee and Client.
- Supertype: Employee and Subtypes: Management, Sales person and Receptionist.

Weak Entities:

- Management – (Employee_Number [PK, FK], Report_ID[FK], Report).
- Sales Person – (Employee_Number [PK, FK], Client_Number[FK]).
- Receptionist – (Employee_Number [PK, FK], Scheduled_Appointment).

Binary Relationships:

- Binary relationships where all mapped out where two entities/substances take part and participate, and is the foremost common relationship degree.

Higher-degree Relationships:

- Higher-degree relationships were also mapped out where more than three entities or substances take part and participate.
- A relationship between a Bank and Financial Department, where a Bank places a request to one and only one Financial department and Financial Department provides a request on one and only one Bank is regarded as a higher-degree relationship.

Validate logical model integrity constraints

Validation:

PERSON

- Person_ID
  - Is a valid person ID
  - Type: Number
  - Range: low value: 0000000000000
  - High value: 9999999999999
  - Display format: 9999999999999
  - Length: 13

- FirstName
  - Is a valid Person's FirstName
  - Type: Character
  - Display format: XXXXXXXXXX….
  - Length:50

- LastName
  - Is a valid Person's LastName
  - Type: Character
  - Display format: XXXXXXXXXX…..
  - Length:50
  -
- PhoneNumber I
  - Is a valid Person's PhoneNumber
  - Type: Numeric
  - Range: low value = +27 0000000
  - High value: +27 999999999
  - Display format:
  - Length: 11

- Email
  - Is a valid Person's Email address
  - Type: character
  - Display format: XXXXXXXXX…..@XXXXXXXXXX.XXX
  - Length: 100

- PersonType
  - Is a valid type
  - Type: character
  - Display format: XXX
  - Length: unlimited
- Username
  - Is a valid Person's Username
  - Type: character
  - Display format: XXXXXXXXXXXXXXX……
  - Length: 100
- Password
  - Is a valid Person's Password
  - Type: character
  - Display format: XXXXXXXXXX
  - Length: 100
- Active_user
  - Is a valid Person Active status
  - Type: character
  - Display format: X
  - Length: 1

CLIENT

- Person_ID
  - Is a valid Client's ID and references a Person record
  - Type: Numeric
  - Range: low value: 0000000000000
  - High value: 9999999999999
  - Display format: 9999999999999
  - Length: 13

- FirstName
  - Is a valid Client's FirstName
  - Type: Character
  - Display format: XXXXXXXXXX…..
  - Length:50

- LastName
  - Is a valid Client's LastName
  - Type: Character
  - Display format: XXXXXXXXXX…..
  - -Length:50
- PhoneNumber

  - Is a valid Client’s PhoneNumber
  - Type: Numeric
  - Range: low value = +27 0000000
  - High value: +27 999999999
  - Display format: +27 99 999 9999
  - Length: 11

- Email
  - Is a valid Client’s Email address
  - Type: character
  - Display format: XXXXXXXXX…..@XXXXXXXXXX.XXX
  - Length: 100

- PersonType
  - Is a valid Client’s type
  - Type: character
  - Display format: XXX
  - Length: unlimited

- Username
  - Is a valid Client’s Username
  - Type: character
  - Display format: XXXXXXXXXXXXXXX……
  - Length: 100

- Password
  - Is a valid Client’s Password
  - Type: character
  - Display format: XXXXXXXXXX
  - Length: 100

- Active_user
  - Is a valid Client’s Active status
  - Type: character
  - Display format: X
  - Length: 1
- Salesperson
  - Is a valid Client’s salesperson Type
  - Type: character
  - Display format: X
  - Length: 1

FORM

- Form_ID
  - Is a valid Form_ID
  - Type: numeric
  - Range: low value: 00000000000 high value: 9999999999
  - Display format: 99999999999
  - Length:10

- FirstName
  - Is a valid Form’s FirstName
  - Type: Character
  - Display format: XXXXXXXXXX….
  - Length:50

- LastName
  - Is a valid Form’s LastName
  - Type: Character
  - Display format: XXXXXXXXXX….
  - Length:50

- ID_Number
  - Is valid Form’s ID_Number
  - Type: numeric
  - Range: low value: 0000000000000 high value: 9999999999999
  - Display format: 9999999999999
  - Length:13

- Car_ID
  - Is valid Form’s Car_ID
  - Type: character
  - Display format: XXX XXX XX
  - Length:10

- Signature
  - Is a valid Form’s Signature
  - Type: character
  - Display format: XXXXXXXXXXXXXXXXXXXX
  - Length:25

EMPLOYEE

- Person_ID
    - Is a valid Employee’s ID and references a Person record
    - Type: Numeric
    - Range: low value: 0000000000000
    - High value: 9999999999999
    - Display format: 9999999999999
    - Length: 13

- FirstName
    - Is a valid Employee’s FirstName
    - Type: Character
    - Display format: XXXXXXXXXX….
    - Length:50
- LastName
    - Is a valid Employee’s LastName
    - Type: Character
    - Display format: XXXXXXXXXX….
    - Length:50

- PhoneNumber
    - Is a valid Employee’s PhoneNumber
    - Type: Numeric
    - Range: low value = +27 0000000
    - High value: +27 999999999
    - Display format:
    - Length: 11
- Email
    - Is a valid Employee’s Email address
    - Type: character
    - Display format: XXXXXXXXX….@XXXXXXXXXX.XXX
    - Length: 100
- PersonType
    - Is a valid Employee’s type
    - Type: character
    - Display format: XXX
    - Length: unlimited

- Username
    - Is a valid Employee’s Username
    - Type: character
    - Display format: XXXXXXXXXXXXXXX……
    - Length: 100

- Password
    - Is a valid Employee’s Password
    - Type: character
    - Display format: XXXXXXXXXX
    - Length: 100

- Active_user
    - Is a valid Employee’s Active status
    - Type: character
    - Display format: X
    - Length: 1

- Position_Title
    - Is a valid Employee’s full job Title
    - Type: character
    - Display format:
    - Length: 100

- Position_Type
    - Is a valid Employee’s type is optional
    - Type: character
    - Display format: X
    - Valid entries: T, null
    - Length: 1

- Tel_num
    - Is a valid Employee’s Telephone Number
    - Type: numeric
    - Display format: +27999999999
    - Length: 11

DEPARTMENT

- Person_ID
    - Is a valid Department ID and references a Person record
    - Type: Numeric
    - Range: low value: 0000000000000
    - High value: 9999999999999
    - Display format: 9999999999999
    - Length: 13

- FirstName
    - Is a valid Department FirstName
    - Type: Character
    - Display format: XXXXXXXXXX….
    - Length:50

- LastName
    - Is a valid Department LastName
    - Type: Character
    - Display format: XXXXXXXXXX….
    - Length:50

- PhoneNumber
    - Is a valid Department PhoneNumber
    - Type: Numeric
    - Range: low value = +27 0000000
    - High value: +27 999999999
    - Display format: +27 999999999
    - Length: 11

- Email
    - Is a valid Department Email address
    - Type: character
    - Display format: XXXXXXXXX….@XXXXXXXXXX.XXX
    - Length: 100

- PersonType
    - Is a valid Department type
    - Type: character
    - Display format: XXX
    - Length: unlimited

- Username
    - Is a valid Department Username
    - Type: character
    - Display format: XXXXXXXXXXXXXXX……
    - Length: 100

- Password
    - Is a valid Department Password
    - Type: character
    - Display format: XXXXXXXXXX

- Length: 100

- Active_user
  - Is a valid Department Active status
  - Type: character
  - Display format: X
  - Length: 1

- Postion_Title
  - Is a valid Department full job title.
  - Type: character
  - Display format:
  - Length: 100
  -
- Position_Type
  - Is a valid Employee's type is optional
  - Type: character
  - Display format: X
  - Valid entries: T, null
  - Length: 1

- Department_ID
  - Is a valid department_ID
  - Type: numeric
  - Display format: 99999
  - Length: 5
- Description
  - Is a valid description
  - Type: character
  - Display format: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX…. XXXXXXXXX
  - Length: 1000

FINANCIAL_DEPARTMENT

- Department_ID
  - Is a valid department_ID
  - Type: numeric
  - Display format: 99999
  - Length: 5

- Financial_Reports
  - Is a valid report
  - Type: character

  - Display format: XXX
  - Length: unlimited

- Financial_Achives
  - Is a valid Financial_Achives
  - Type: character
  - Display format: XXX
  - Length: unlimited

- Tel_Num
  - Is a valid Financial_department Telephone Number
  - Type: numeric
  - Display format: +27999999999
  - Length: 11

- Email
  - Is a valid Department Email address
  - Type: character
  - Display format: XXXXXXXXX….@XXXXXXXXXX.XXX
  - Length: 100

BANK

- Bank_ID
  - Is a valid bank ID
  - Type: numeric
  - Length: 13
  -
- Department_ID
  - Is a valid department_ID
  - Type: numeric
  - Display format: 99999
  - Length: 5

- Bank_report
  - Is a valid report
  - Type: character
  - Display format: XXX
  - Length: unlimited

- Bank_statement
  - Is a valid statement
  - Type: character

- Display format: XXX
- Length: unlimited

- Bank_telephone
  - Is a valid telephone
  - Type: numeric
  - Range: low value = 0 high value = 9
  - Display format: 9999 or +9999
  - Length: 10

MANAGEMENT

- Management_ID
  - Is a valid management_ID
  - Type: numeric
  - Range low value:000000000000000000000   high value:99999999999999
  - Display format:999999999999999999
  - Length:20

- Employee_Number
  - Is a valid Employee's Number that reference the Employee record.
  - Type: numeric
  - Range: low value 0000000000000000000 high value:9999999999999999
  - Display format: 9999999999999999999
  - Length: 20

- Report_ID
  - Is a valid report
  - Type: character
  - Display format: XXX
  - Length: unlimited

SALES_PERSON

- Employee_number
  - Is a valid Employee's Number that reference the Employee record.
  - Type: numeric
  - Range: low value 0000000000000000000 high value:999999999999999
  - Display format: 9999999999999999999
  - Length: 20

- Client_Number
    - Is a valid Client number that reference the Client record
    - Type: numeric
    - Range: low value: 00000000000000 high value: 99999999999999
    - Display format:9999999999999999
    - Length:13

RECEPTIONIST

- Employee_Number
    - Is a valid Employee’s Number that reference the Employee record
    - Type: numeric
    - Range: low value 00000000000000000000 high value:99999999999999999
    - Display format: 99999999999999999999
    - Length: 20

- Scheduled_Appointment
    - Is a valid schedule_appointment
    - Type: Time
    - Display format:
    - Length: 20

REPORT

- Report_ID
    - Is valid Report_ID
    - Type: numeric
    - Range: low value: 0000000000000000000000 high value: 99999999999999999
    - Display format: 9999999999999999999999999999999999999999
    - Length:50

- Employee_number
    - Is a valid Employee’s Number that reference the Employee record
    - Type: numeric
    - Range: low value 00000000000000000000 high value:9999999999999999
    - Display format: 99999999999999999999

- Length: 20

CAR

- Car_ID
  - Is valid Car_ID
  - Type: character
  - Display format: XXX XXX XX
  - Length:10
- Car_Name
  - Is valid Car_Name
  - Type: character
  - Display format: XXX XX
  - Length:50
  -
- Car_TypeModel
  - Is valid Car_TypeModel
  - Type: character
  - Display format: XXXXXXXXXXXXX
  - Length:50

- Engine
  - Is valid Engine
  - Type: character
  - Display format: XXXXXXXXXXXXX
  - Length:50

- Year
  - Is valid Year
  - Type: numeric
  - Range: low value: 0000 high value: 9999
  - Display format: 9999
  - Length:4

- DriversCode
  - Is valid DriversCode
  - Type: numeric
  - Range: low value: 00 high value: 99
  - Display format: 9999
  - Length:2

- Quantity
  - Is valid Quantity

    - Type: numeric
    - Range: low value: 0000 high value: 9999
    - Display format: 9999
    - Length:4

REQUEST_PROCESS

- Request_code
    - Is a valid Request_process Password
    - Type: character
    - Display format: XXXXXXXXXX
    - Length: 100

- Username
    - Is a valid Request_process Username
    - Type: character
    - Display format: XXXXXXXXXXXXXXX……
    - Length: 100

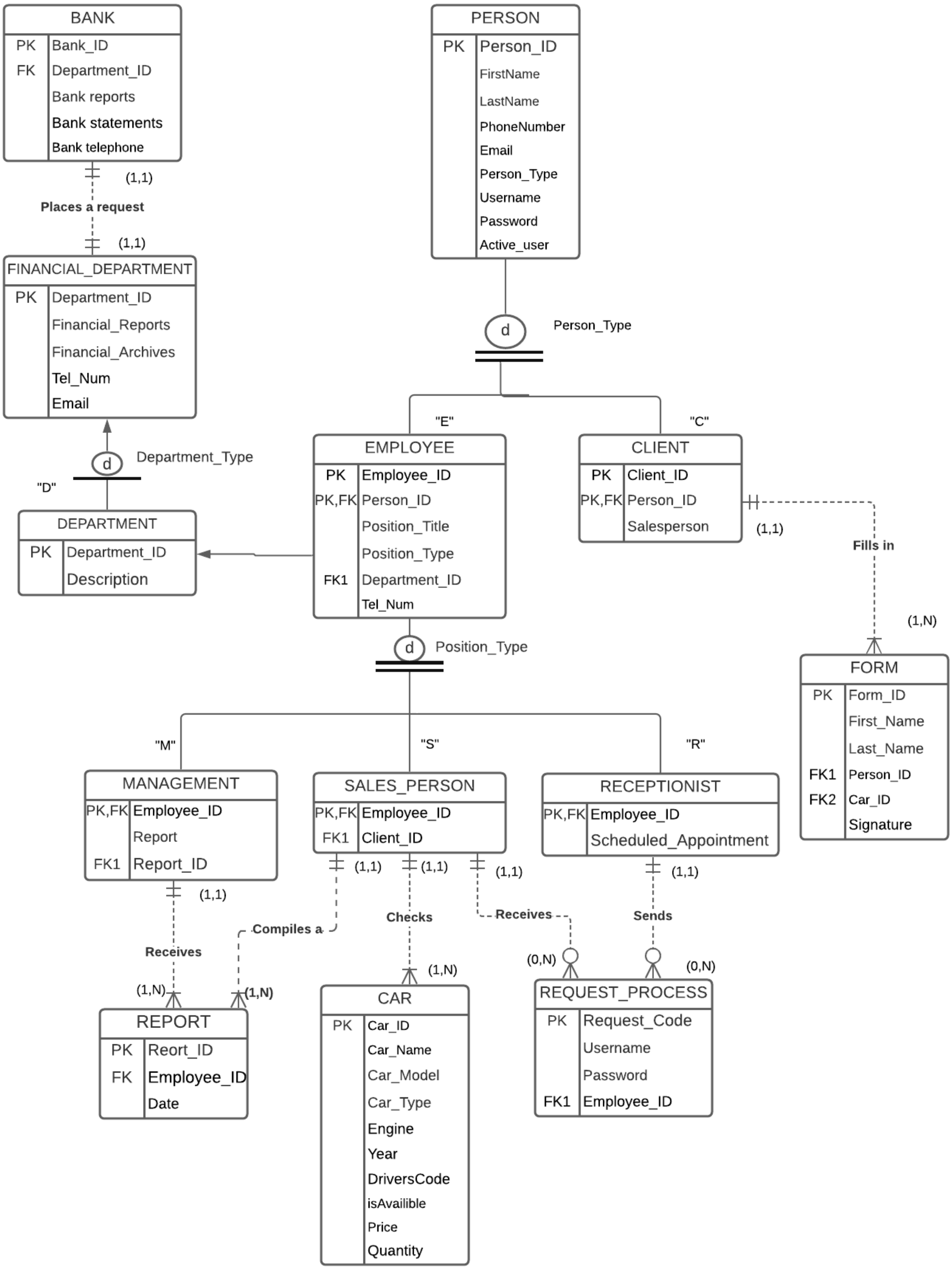
- Password
    - Is a valid Department Password
    - Type: character
    - Display format: XXXXXXXXXX
    - Length: 100

- Employee_Number
    - Is a valid Employee’s Number that reference the Employee record
    - Type: numeric
    - Range: low value 00000000000000000000 high value:9999999999999999
    - Display format: 99999999999999999999
    - Length: 20

PHASE 3 – PHYSICAL DESIGN

Database Design/Layout

The ERD model below is the depiction of the database construction. It will be used for developing sql statements in the Oracle sql developer.

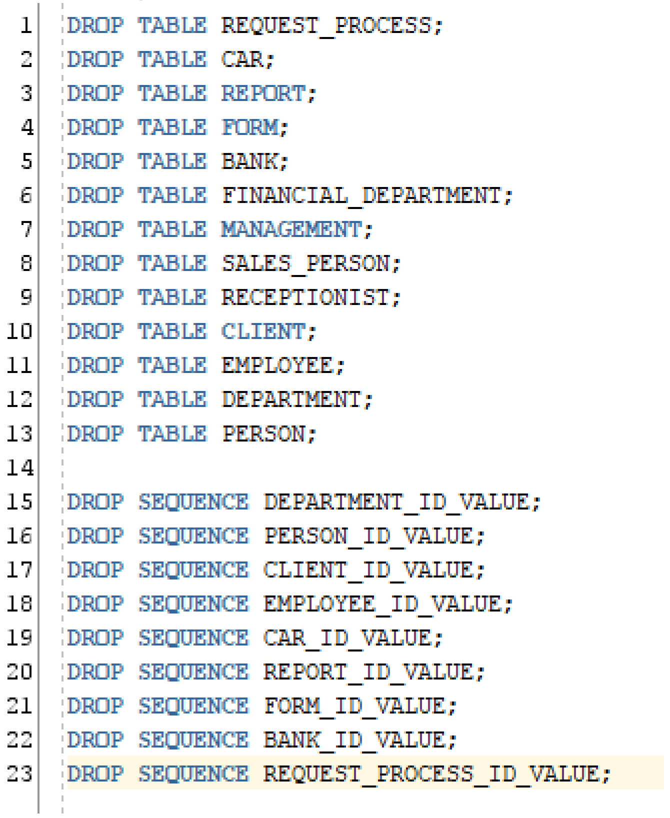


Database Objects

In this section, we will be constructing the initial database as well as determining all relationships between the non-identical tables in the database. When the database construction is complete, the database will then be populated with default records that will be utilized to exhibit the queries that will be disclosed in section 3 of this document.

Removing All Tables to Allow the Creation of New Tables

In this sub-section, we will be demonstrating relevant SQL statements that will be used to drop/delete all the tables that share same name as the tables we are trying to create. This means that we are preventing chances of the existence of duplicate tables. The SQL statements will automatically remove all the tables with its stored records. The order of dropping/deleting the tables is "children entities" first then followed by the "parent entities", this important because we cannot drop/delete a table that foreign keys in other tables. It is important that we drop the tables first if we want to drop sequences, only then we can drop the sequences to ensure the non-existence of errors during the recreation of tables. Unlike sequences, views can be simply dropped without initiating any negative effects. Therefore, it is recommended that tables should not be dropped after they have been created as this could result in loss of data that could cause major issues when tables are created again.



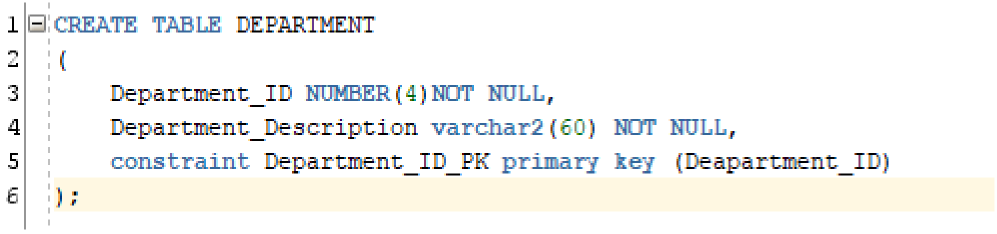
```sql
DROP TABLE REQUEST_PROCESS;
DROP TABLE CAR;
DROP TABLE REPORT;
DROP TABLE FORM;
DROP TABLE BANK;
DROP TABLE FINANCIAL_DEPARTMENT;
DROP TABLE MANAGEMENT;
DROP TABLE SALES_PERSON;
DROP TABLE RECEPTIONIST;
DROP TABLE CLIENT;
DROP TABLE EMPLOYEE;
DROP TABLE DEPARTMENT;
DROP TABLE PERSON;

DROP SEQUENCE DEPARTMENT_ID_VALUE;
DROP SEQUENCE PERSON_ID_VALUE;
DROP SEQUENCE CLIENT_ID_VALUE;
DROP SEQUENCE EMPLOYEE_ID_VALUE;
DROP SEQUENCE CAR_ID_VALUE;
DROP SEQUENCE REPORT_ID_VALUE;
DROP SEQUENCE FORM_ID_VALUE;
DROP SEQUENCE BANK_ID_VALUE;
DROP SEQUENCE REQUEST_PROCESS_ID_VALUE;
```

Table Creation

In table creation, we introduce statements that will be used to create tables within our database. We will be assigning primary key, placing foreign keys to their respecting tables and assigning the correct data types of all the fields. In most cases, primary keys and foreign keys will be assigned number data type and any text related field will receive the varchar2 data type and the maximum size of that specific field will be indicated in brackets next to the varchar2 assignment. The date related fields will be assigned the Date data type. Addition exceptions will be mentioned in the tables in their respective sections. A number with two or more parameters specifies precision and the scale of that particular field.

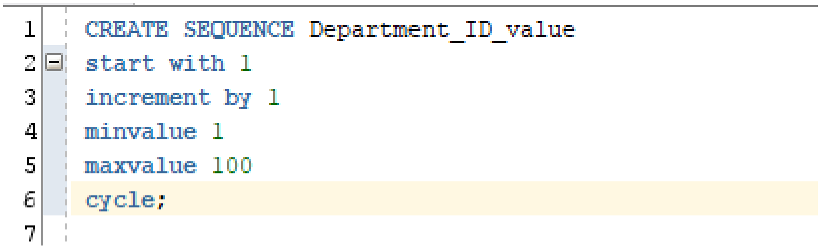
- **Department Table**

The department table stores information about various departments within the car dealership. It consists of two fields which are, Department_ID and Description. The Department _ID is used as the Primary Key to Uniquely identify each department, the Description stores a brief description of the department identified with a maximum size of characters. Both values cannot be left null, meaning that they contain the NOT NULL CONSTRAINT. This table has no foreign keys.



```
CREATE TABLE DEPARTMENT
(
    Department_ID NUMBER(4)NOT NULL,
    Department_Description varchar2(60) NOT NULL,
    constraint Department_ID_PK primary key (Deapartment_ID)
);
```

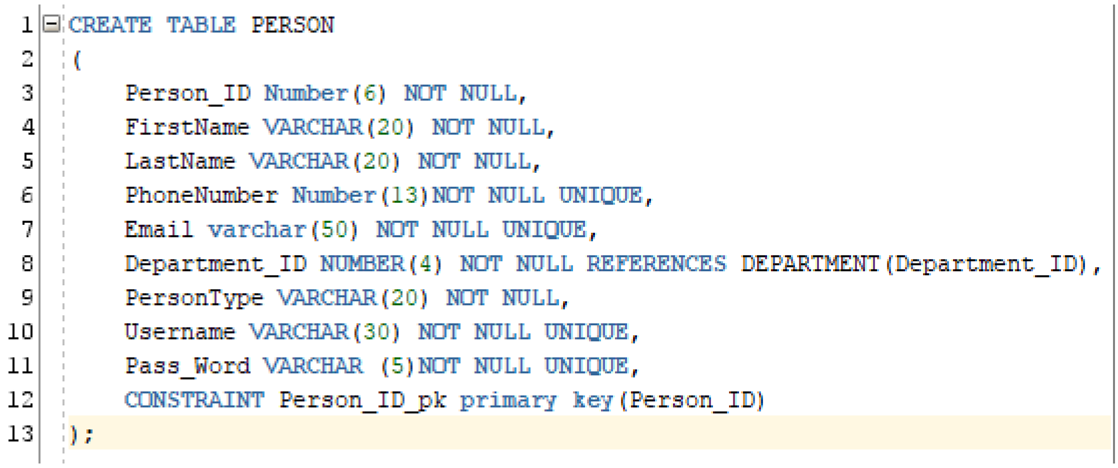
The above sequence will be displayed alongside this table and will be used to determine the value of Department_ID. Sequences help the to minimize the risk of entering an invalid value into the primary slot. This ensure data integrity is maintained and uniqueness of each record. The cycle keyword indicates that the maximum value is reached and the sequence must start again at the minimum value. However, this could cause chances of the appearance of duplicate primary keys which is an issue when the Department_ID gets near the maximum value.



```
CREATE SEQUENCE Department_ID_value
start with 1
increment by 1
minvalue 1
maxvalue 100
cycle;
```

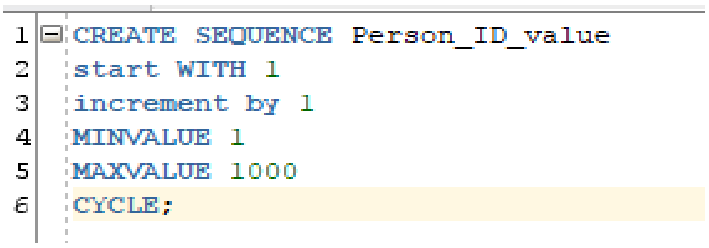
- **Person Table**

The person table consists of general information about employee people and people who are clients. This is information such as their name, last name, email address and so on as indicated on the design layout. A unique primary key ID will be received by the Person table which is Person_ID. All the fields except the email address field have the NOT NULL restriction and the standard datatypes have been implemented. On that note, the phone number and email address field hold the UNIQUE constraint simply meaning that these fields require unique values for every records in the table. The big change is that in this table is the addition of a sub-type discriminator namely the Person_Type field. The person table is the super-type with 2 sub-types; namely Employee and Client. This means that each and every person must belong to one and only one of these sub-types. The sub-type discriminator will determine which sub-type this field belongs to by using "E" for Employee, and "C" for Client and so on. More on this will be explained in the upcoming sections.



```
CREATE TABLE PERSON
(
    Person_ID Number(6) NOT NULL,
    FirstName VARCHAR(20) NOT NULL,
    LastName VARCHAR(20) NOT NULL,
    PhoneNumber Number(13)NOT NULL UNIQUE,
    Email varchar(50) NOT NULL UNIQUE,
    Department_ID NUMBER(4) NOT NULL REFERENCES DEPARTMENT(Department_ID),
    PersonType VARCHAR(20) NOT NULL,
    Username VARCHAR(30) NOT NULL UNIQUE,
    Pass_Word VARCHAR (5)NOT NULL UNIQUE,
    CONSTRAINT Person_ID_pk primary key(Person_ID)
);
```

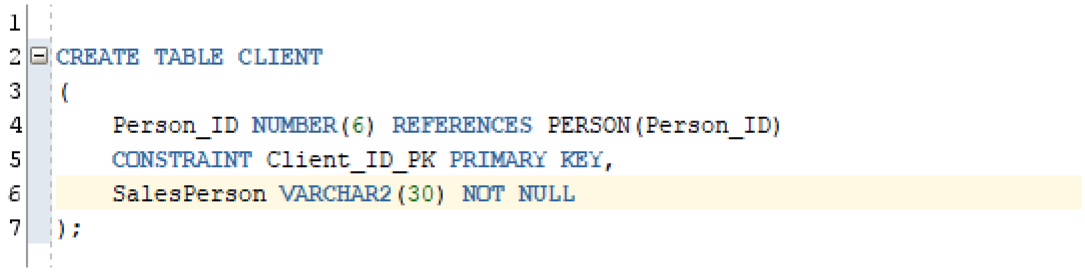
This table has a unique ID that needs a sequence to calculate the Primary Key to ensure it is always valid. No user input is required to determine the Person_ID.



```
CREATE SEQUENCE Person_ID_value
start WITH 1
increment by 1
MINVALUE 1
MAXVALUE 1000
CYCLE;
```

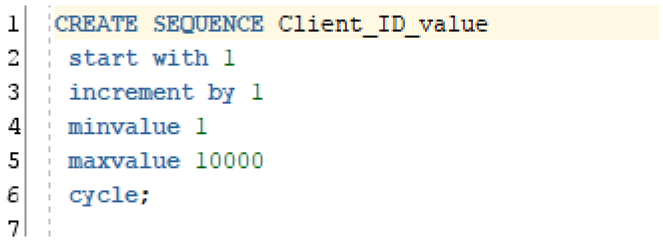
- **Client Table**

The client table consists of all information pertaining the clients only. The Client table is a sub-type of the Person table. Thus the table will inherit the Primary key from the Person table. By doing so, the table indirectly inherits all the attributes from Person table. The Client table has Person_ID as a Primary key and is at the same time a foreign key connecting with the Person table. All of the fields in this table have the NOT NULL constraint meaning that they cannot be left empty as they are mandatory. The SalesPerson field is required to show which client is assigned to which sales person.



```

CREATE TABLE CLIENT
(
    Person_ID NUMBER(6) REFERENCES PERSON(Person_ID)
    CONSTRAINT Client_ID_PK PRIMARY KEY,
    SalesPerson VARCHAR2(30) NOT NULL
);
```

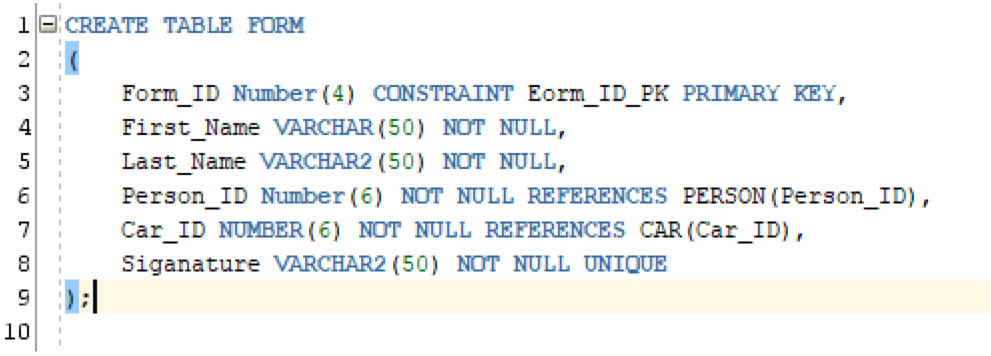
Like many other previous tables, this table has a unique ID that needs a sequence to calculate the Primary Key to ensure it is always valid. No user input is required to determine the Client_ID.



```
CREATE SEQUENCE Client_ID_value
 start with 1
 increment by 1
 minvalue 1
 maxvalue 10000
 cycle;

```

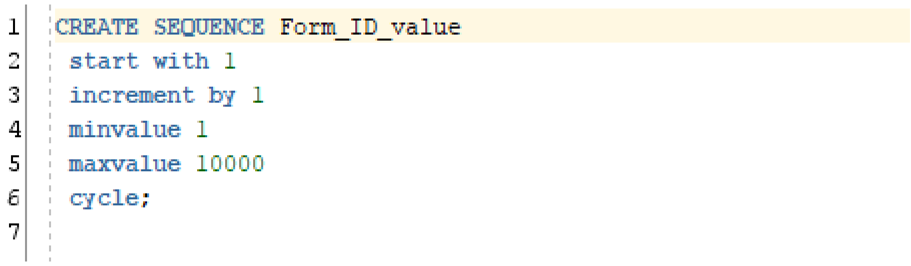
- **Form Table**

The Form table will house all the forms that the car dealership has received from the clients. Other information that will stored in the table is the date that the form was received and signed, the particular client who submitted the form's relevant details and to which sales person is the form sent to details. The department in the car dealership that is responsible for those forms if they are filled accordingly leaving no required field is the reception department. As we know, each form has its assigned unique ID called Form_ID, this will automatically serve as the Primary key. This table has two Foreign keys connecting it to other tables, the Foreign keys are Person_ID and Car_ID. Person_ID refers to the particular client that is giving their personal information on the form, and Car_ID refers to the connection to the type of vehicle the client wants to purchase. All of the information of the fields in this table is required and must not be left null. It has the NOT NULL constraint and the required data type standards have been given, but the signature can simply be the surname and initials of the client.



```
CREATE TABLE FORM
(
    Form_ID Number(4) CONSTRAINT Eorm_ID_PK PRIMARY KEY,
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR2(50) NOT NULL,
    Person_ID Number(6) NOT NULL REFERENCES PERSON(Person_ID),
    Car_ID NUMBER(6) NOT NULL REFERENCES CAR(Car_ID),
    Siganature VARCHAR2(50) NOT NULL UNIQUE
);
```

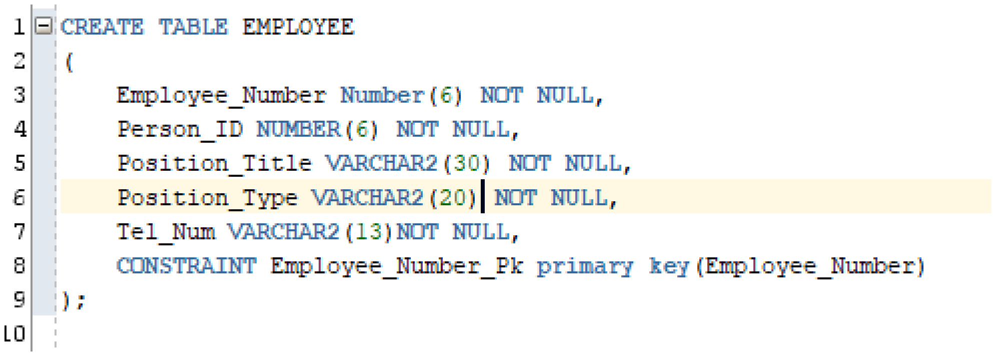
The Unique sequence will ensure that each record has its own unique value without the user having to calculate a valid Primary Key.



```
CREATE SEQUENCE Form_ID_value
 start with 1
 increment by 1
 minvalue 1
 maxvalue 10000
 cycle;
```

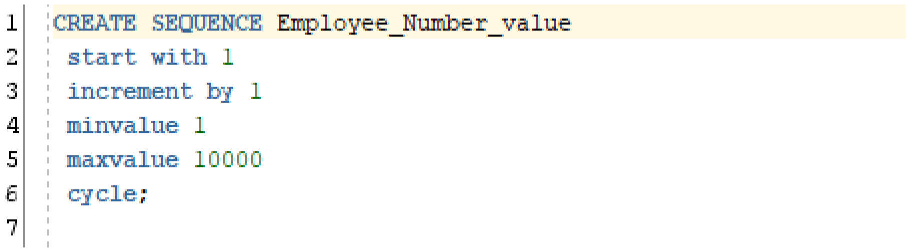
- **Employee Table**

This table consists of many relationships to be addressed, again this table a super-type that will hold general information of the type of employees we consider, namely Position_type, Position_title and the telephone number of the employee's office. A Unique ID (Employee_Number) is assigned to every employee that will also become the Primary key. There is a foreign key housed by this table, identified as Department_ID, this indicates the department in which the employees belong to. All the fields have the NOT NULL constraint and the relevant data standards are implemented. The Tel_no also have a unique constraint meaning that these fields need to be unique for every record in the table. The implemented sub-type discriminator namely, Employee_type field. As the supertype table, the Employee Table has Three sub-types namely, Receptionist, Sales Person and Management. Each employee must belong to only one of these sub-types, thus we have one sub-type discriminator. This is used to demonstrate to which sub-type this field is assigned by making use of "R" for Receptionist, "S" Sales Person and "M" for Management in this case.



```
CREATE TABLE EMPLOYEE
(
    Employee_Number Number(6) NOT NULL,
    Person_ID NUMBER(6) NOT NULL,
    Position_Title VARCHAR2(30) NOT NULL,
    Position_Type VARCHAR2(20) NOT NULL,
    Tel_Num VARCHAR2(13)NOT NULL,
    CONSTRAINT Employee_Number_Pk primary key(Employee_Number)
);
```

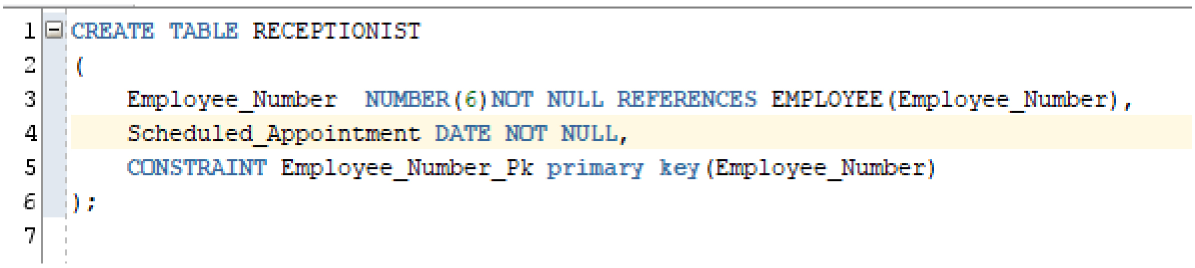
The Employee table also has a unique ID so we also created a Sequence for it. This simply makes it easier for the system and user to select a Primary key for the table.



```
CREATE SEQUENCE Employee_Number_value
 start with 1
 increment by 1
 minvalue 1
 maxvalue 10000
 cycle;
```

- ## Reception Table

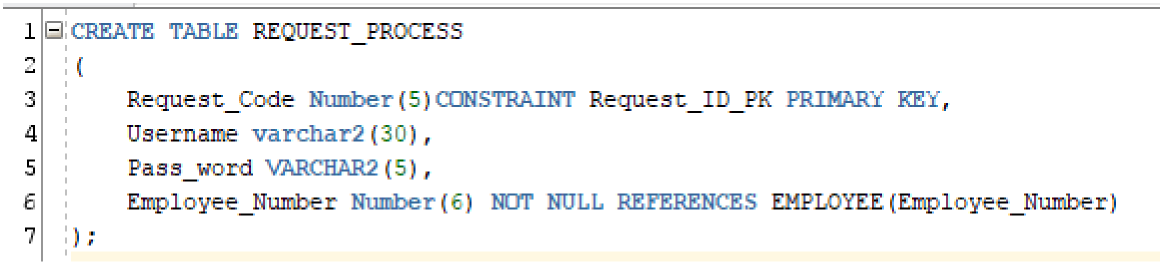
Like all the other sub-types, the Receptionist table inherits its Primary key from the Employee table (Employee _ID), and indirectly inherits all the other attributes of the table such as position type and position title and so on. The Receptionist table contains the scheduled appointment that they use to keep track of appointments with clients on a daily basis. The date values of the schedules will be stored using the Date value character from Oracle. The table has an inherited Primary key which also functions as the foreign key for connection purposes of the tables, and no other foreign keys. All the attributes have a restriction of NOT NULL and follow standard data type values.



```
CREATE TABLE RECEPTIONIST
(
    Employee_Number   NUMBER(6)NOT NULL REFERENCES EMPLOYEE(Employee_Number),
    Scheduled_Appointment DATE NOT NULL,
    CONSTRAINT Employee_Number_Pk primary key(Employee_Number)
);
```

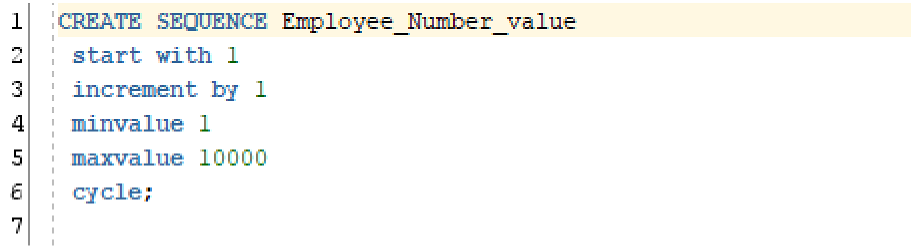
- ## Request Process Table

The purpose of this table is to provide information concerning the process of accessing information about a customer and other private information of the car dealership by the receptionist. Information stored in this table is the username and password of the receptionist for authorization purposes. Each request has a unique ID which is the Primary key of the table called Request_code and it has a single foreign key namely Employee_Number which denotes which employee is responsible for attending which Client. The NOT NULL constraint is applicable to all the fields along with standard data type standards to be used.



```
CREATE TABLE REQUEST_PROCESS
(
    Request_Code Number(5)CONSTRAINT Request_ID_PK PRIMARY KEY,
    Username varchar2(30),
    Pass_word VARCHAR2(5),
    Employee_Number Number(6) NOT NULL REFERENCES EMPLOYEE(Employee_Number)
);
```

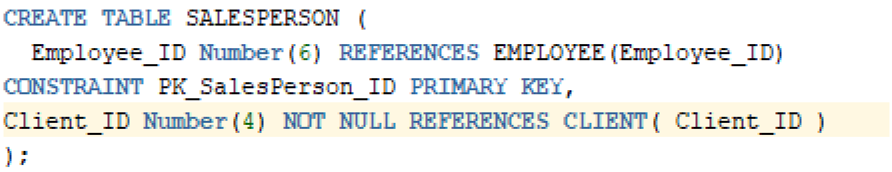
Similarly, to many of our previous tables this table will make uses Unique ID sequence to assign values to the Request_Code primary key. This ensures that the Request_Code will always have a unique value and the system will automatically increment the sequence value without the user having to manually assign the Primary keys.



```
CREATE SEQUENCE Employee_Number_value
 start with 1
 increment by 1
 minvalue 1
 maxvalue 10000
 cycle;
```

- ## Sales Person Table

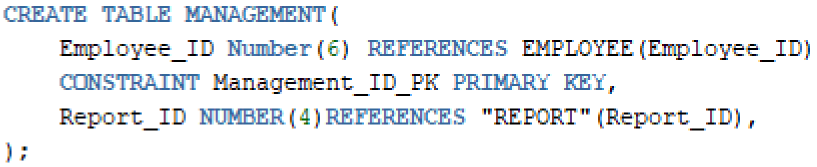
The Sales table is a subtype of the Employee table and inherits its primary key from the Employee table as well the attributes contained in the Employee table. It has a foreign key named Client_ID meaning that the Sales Person is responsible for serving a Client. There is a NOT NULL Constraint for all of the attributes and the standard data type standards is followed by all of the attributes. There are no sequences since there is no UNIQUE_ID.



```
CREATE TABLE SALESPERSON (
  Employee_ID Number(6) REFERENCES EMPLOYEE(Employee_ID)
CONSTRAINT PK_SalesPerson_ID PRIMARY KEY,
Client_ID Number(4) NOT NULL REFERENCES CLIENT( Client_ID )
);
```

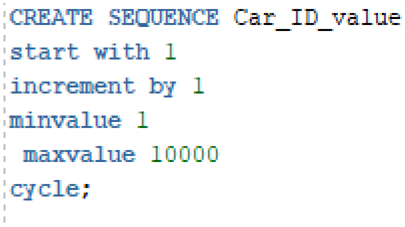
- ## Car Table

The Car table contains all relevant details about the car. It provides the features of the car to be purchased. It has a unique ID that become its primary key. The table has no foreign keys and follows the standard data type standards. The NOT NULL Constraint is applicable to all fields in this table.



```
CREATE TABLE MANAGEMENT(
    Employee_ID Number(6) REFERENCES EMPLOYEE(Employee_ID)
    CONSTRAINT Management_ID_PK PRIMARY KEY,
    Report_ID NUMBER(4)REFERENCES "REPORT"(Report_ID),
);
```

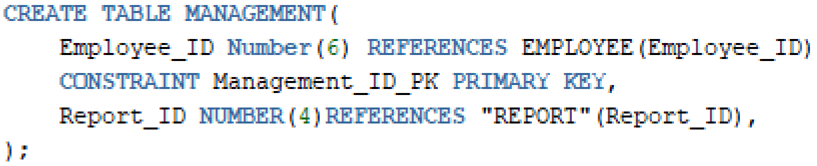
A sequence is required to ensure that the primary key of the table remains unique in every row.



```
CREATE SEQUENCE Car_ID_value
start with 1
increment by 1
minvalue 1
 maxvalue 10000
cycle;
```

- ## Management Table

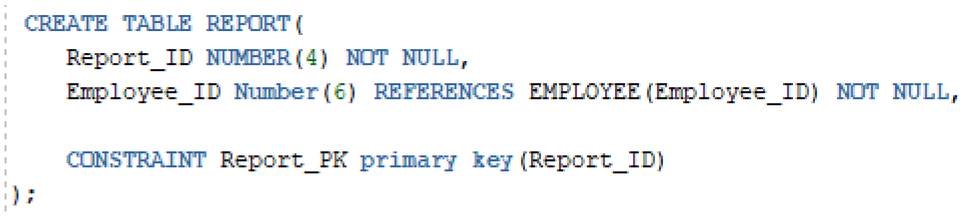
The Management table is a subtype of the Employee table and inherits its primary key from the Employee table as well the attribute contains in the Employee table. It has a foreign key named Report_ID, the management receives a report that is compiled by the salesperson regarding car sales. There is a NOT NULL Constraint for all of the attributes and the standard data type standards is followed by all of the attributes. There are no sequences since there is no UNIQUE_ID. All of the attributes have the NOT NULL constraint.



```
CREATE TABLE MANAGEMENT(
    Employee_ID Number(6) REFERENCES EMPLOYEE(Employee_ID)
    CONSTRAINT Management_ID_PK PRIMARY KEY,
    Report_ID NUMBER(4)REFERENCES "REPORT"(Report_ID),
);
```

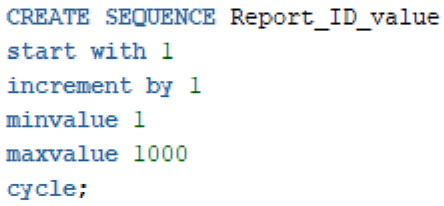
- **Report Table**

  The Report table will be used to store information about the client's purchases, number of cars sold out, remaining cars in the precinct and number of cars brought into the company monthly. It has a unique ID that become its primary key and has a foreign key Employee_ID which denotes that an employee generates a report required by the management. All of the attributes have the NOT NULL constraint.



```
CREATE TABLE REPORT(
   Report_ID NUMBER(4) NOT NULL,
   Employee_ID Number(6) REFERENCES EMPLOYEE(Employee_ID) NOT NULL,

   CONSTRAINT Report_PK primary key(Report_ID)
);
```

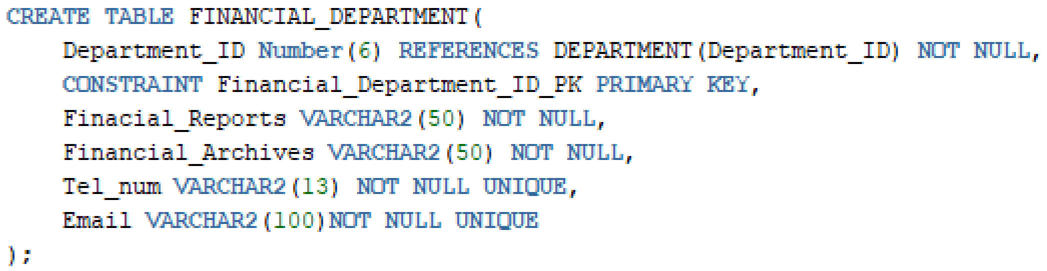
A sequence will be required to ensure that the primary key of the table remains unique in every row so there won't be a need for a user to provide input.



```
CREATE SEQUENCE Report_ID_value
start with 1
increment by 1
minvalue 1
maxvalue 1000
cycle;
```

- **Financial Department Table**

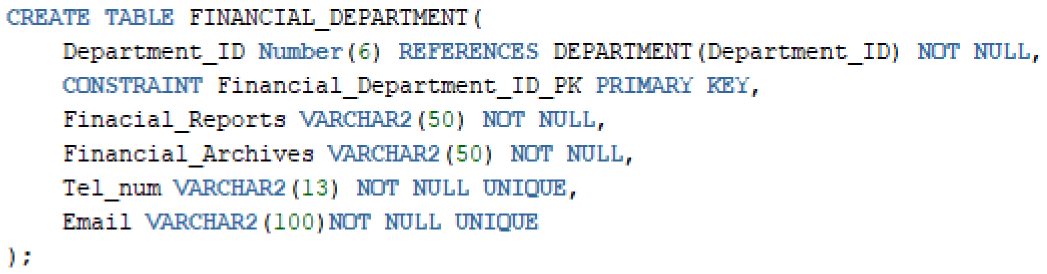
  The Financial department table contains all the information about the finances of the company. It is a subtype of the Department table and inherits its primary key from the Department table as well the attributes contained in the Department table. All of the attributes have the NOT NULL constraint.



```
CREATE TABLE FINANCIAL_DEPARTMENT(
    Department_ID Number(6) REFERENCES DEPARTMENT(Department_ID) NOT NULL,
    CONSTRAINT Financial_Department_ID_PK PRIMARY KEY,
    Finacial_Reports VARCHAR2(50) NOT NULL,
    Financial_Archives VARCHAR2(50) NOT NULL,
    Tel_num VARCHAR2(13) NOT NULL UNIQUE,
    Email VARCHAR2(100)NOT NULL UNIQUE
);
```

- **Bank Table**

The Financial department table contains all the information about the finances of the company. It is a subtype of the Department table and inherits its primary key from the Department table as well the attributes contained in the Department table. All of the attributes have the NOT NULL constraint.



```
CREATE TABLE FINANCIAL_DEPARTMENT(
    Department_ID Number(6) REFERENCES DEPARTMENT(Department_ID) NOT NULL,
    CONSTRAINT Financial_Department_ID_PK PRIMARY KEY,
    Finacial_Reports VARCHAR2(50) NOT NULL,
    Financial_Archives VARCHAR2(50) NOT NULL,
    Tel_num VARCHAR2(13) NOT NULL UNIQUE,
    Email VARCHAR2(100)NOT NULL UNIQUE
);
```

We have now completed table creation for all tables, we subsequently no have to deal with populating those tables in the consecutive section.

Populating the Database

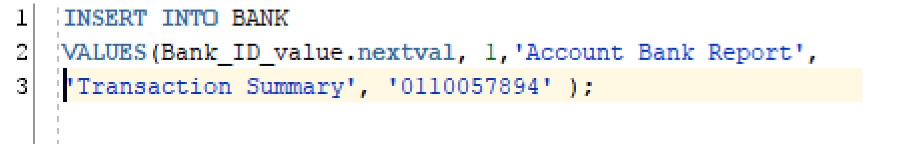
We are informed in section 2.2 during table creation that each field has a specific data type; this is important and depends on the data that needs to be stored in that attribute. These are the data types to be used:

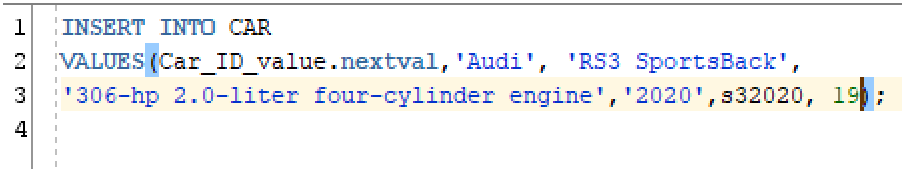
- An Auto Number field where we can make use of a Number field with a sequence to automatically assign values to that field.
- Number which will be used for any fields with numerical values.
- Varchar2 which will be used with fields that contain text.
- The Yes/No data type, but because Oracle does not support Yes/No data types, we can replace it by using the Number field with the value 1 denoting a Yes and value 0 denoting a No.
- A Date data type that will placed with fields concerning dates storing.

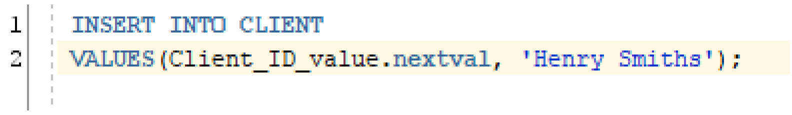
Various methods can be used to simply insert data into the tables that have been created in the previous section. This could be by importing data from a C# program or hardcode the data from scratch into the database. We will be doing the latter by simply creating default records and inserting them into the table.

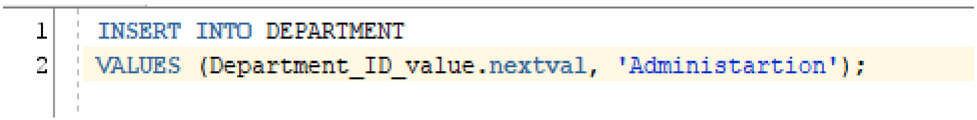
We are not required to specify the attributes after the INSERT INTO statement and can just provide the values, this is because we are inserting a value into every field found in the tables. However, if we need to enter specific values into the table, we would need to specify those fields directly after the table name.
The following consists of our examples:

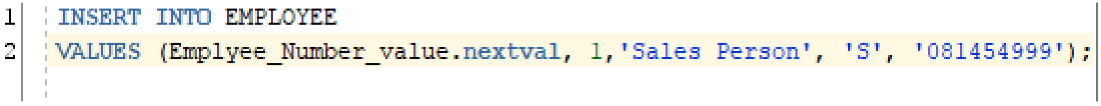
Table Name	INSERT INTO Example

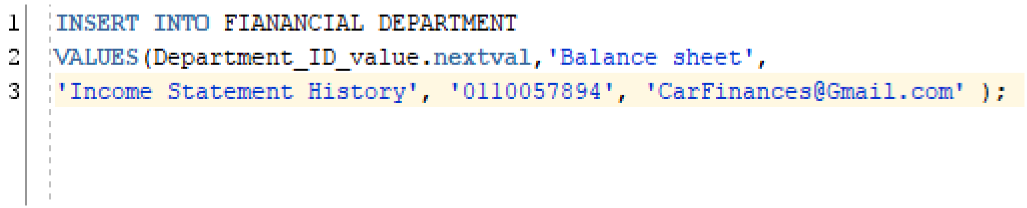
BANK	
	INSERT INTO BANK
	VALUES(Bank_ID_value.nextval, 1,'Account Bank Report',
	'Transaction Summary', '0110057894');

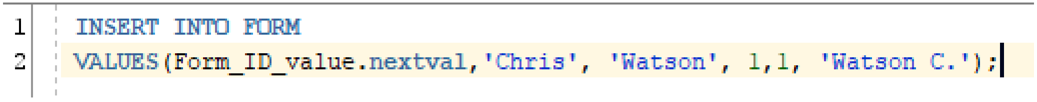
CAR	
	INSERT INTO CAR
	VALUES(Car_ID_value.nextval,'Audi', 'RS3 SportsBack',
	'306-hp 2.0-liter four-cylinder engine','2020',s32020, 19);

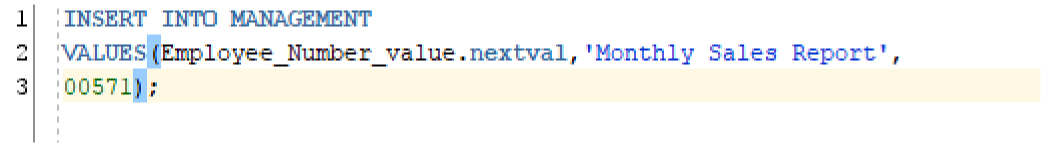
CLIENT	
	INSERT INTO CLIENT
	VALUES(Client_ID_value.nextval, 'Henry Smiths');

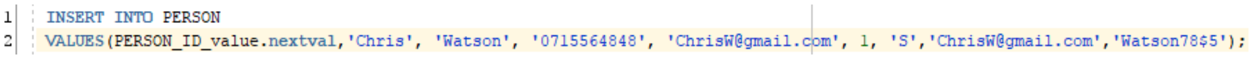
DEPARTMENT	
	INSERT INTO DEPARTMENT
	VALUES (Department_ID_value.nextval, 'Administartion');

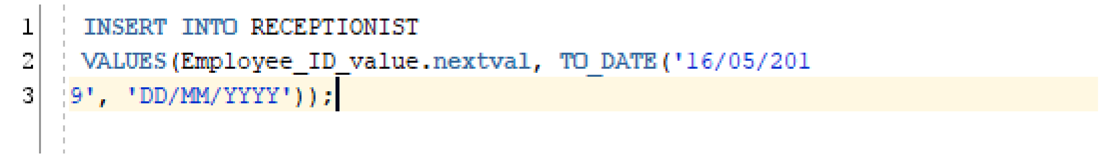
EMPLOYEE	
	INSERT INTO EMPLOYEE
	VALUES (Emplyee_Number_value.nextval, 1,'Sales Person', 'S', '081454999');

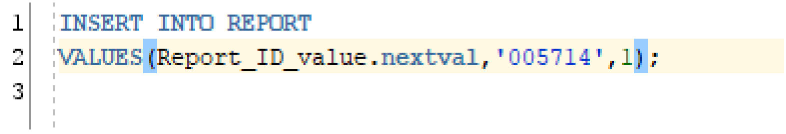
FINANCIAL DEPARTMENT	
	INSERT INTO FIANANCIAL DEPARTMENT
	VALUES(Department_ID_value.nextval,'Balance sheet',
	'Income Statement History', '0110057894', 'CarFinances@Gmail.com');

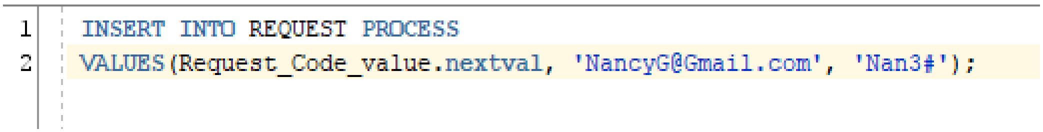
FORM	
	INSERT INTO FORM
	VALUES(Form_ID_value.nextval,'Chris', 'Watson', 1,1, 'Watson C.');

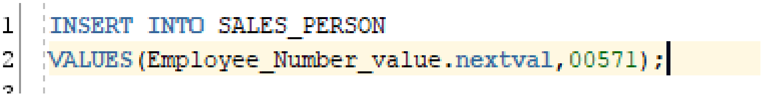
MANAGEMENT	
	INSERT INTO MANAGEMENT
	VALUES(Employee_Number_value.nextval,'Monthly Sales Report',
	00571);

PERSON	
	INSERT INTO PERSON
	VALUES(PERSON_ID_value.nextval,'Chris', 'Watson', '0715564848', 'ChrisW@gmail.com', 1, 'S','ChrisW@gmail.com','Watson78$5');

RECEPTIONIST	
	INSERT INTO RECEPTIONIST
	VALUES(Employee_ID_value.nextval, TO_DATE('16/05/201
	9', 'DD/MM/YYYY'));

REPORT	
	INSERT INTO REPORT
	VALUES(Report_ID_value.nextval,'005714',1);

REQUEST PROCESS	
	INSERT INTO REQUEST PROCESS
	VALUES(Request_Code_value.nextval, 'NancyG@Gmail.com', 'Nan3#');

SALES PERSON	
	INSERT INTO SALES_PERSON
	VALUES(Employee_Number_value.nextval,00571);

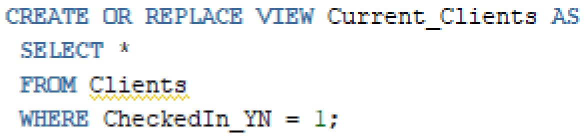
The above mentioned statements are examples of valid insert statements, our code will contain various versions of these insert statements for each of the tables providing them with fata that we can use to test our queries.

Creating Views

The purpose of creating views for our database is to represent subsets of data from the tables. This is done to ease the retrieval of results of complicated queries and to restrict access to data by displaying selected columns from the table.

I. Client view

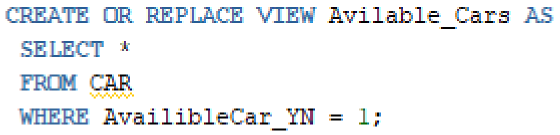
The Client view retrieves the information from Person. Client view shows clients who currently purchasing cars from the dealership and the potential buyers.



```
CREATE OR REPLACE VIEW Current_Clients AS
 SELECT *
 FROM Clients
 WHERE CheckedIn_YN = 1;
```

II. View that contain information about the Cars in the Auto dealership

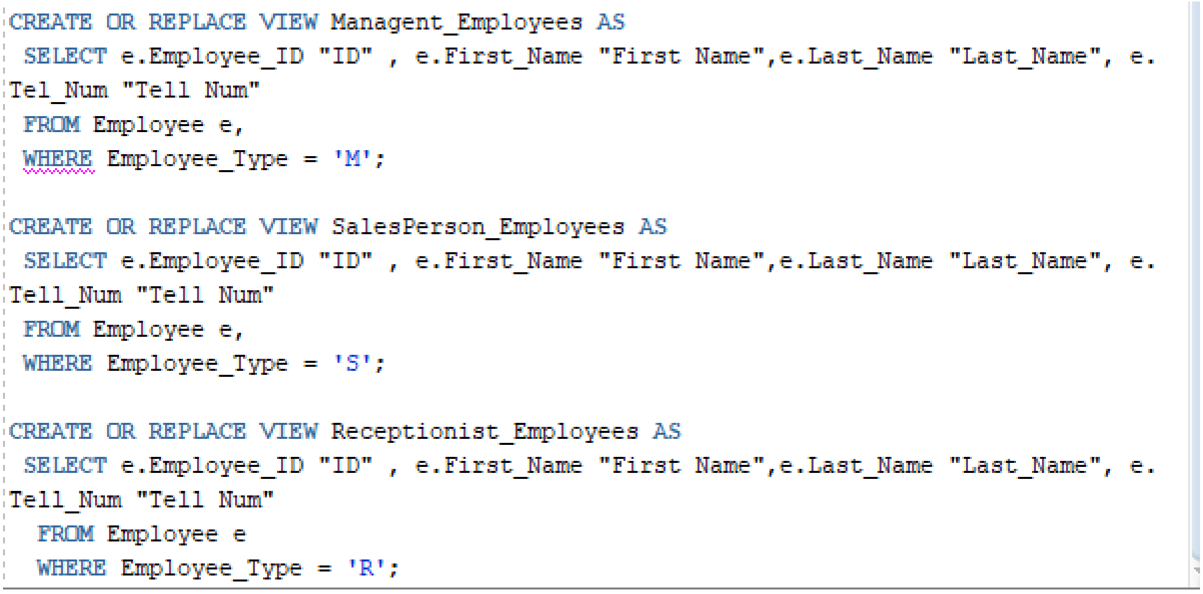
The view about cars retrieves the information about the car and its availability.



```
CREATE OR REPLACE VIEW Avilable_Cars AS
 SELECT *
 FROM CAR
 WHERE AvailibleCar_YN = 1;
```

III. Management, Salesperson and Receptionist views

These views originally retrieve all the information from Person and Employee tables it shows employees who are currently active within the organisation.



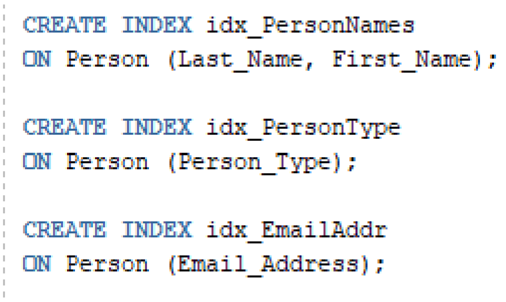
```
CREATE OR REPLACE VIEW Managent_Employees AS
 SELECT e.Employee_ID "ID" , e.First_Name "First Name",e.Last_Name "Last_Name", e.
Tel_Num "Tell Num"
 FROM Employee e,
 WHERE Employee_Type = 'M';

CREATE OR REPLACE VIEW SalesPerson_Employees AS
 SELECT e.Employee_ID "ID" , e.First_Name "First Name",e.Last_Name "Last_Name", e.
Tell_Num "Tell Num"
 FROM Employee e,
 WHERE Employee_Type = 'S';

CREATE OR REPLACE VIEW Receptionist_Employees AS
 SELECT e.Employee_ID "ID" , e.First_Name "First Name",e.Last_Name "Last_Name", e.
Tell_Num "Tell Num"
  FROM Employee e
  WHERE Employee_Type = 'R';
```

Creating Indexes

Creating for our Database will enhance the performance of some queries. This will speed the retrieval of rows by using a pointer.

- **Person Table Indexes**

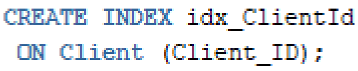
Here indexes will be placed on FirstName, LastName and email address as well as the subtype discriminator(Person_Type), by this a person can be located easily by their type as to whether the is an employee or the person is client.



```
CREATE INDEX idx_PersonNames
ON Person (Last_Name, First_Name);

CREATE INDEX idx_PersonType
ON Person (Person_Type);

CREATE INDEX idx_EmailAddr
ON Person (Email_Address);
```

- **Client Table Indexes**

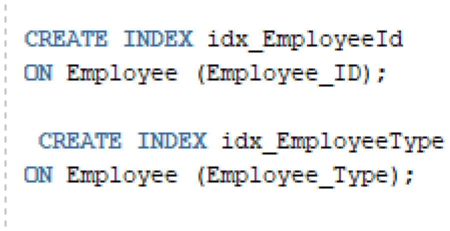
Indexes are placed on the client ID this will help in showing the current clients and it will also help with calculating the number of clients in the system.



```
CREATE INDEX idx_ClientId
 ON Client (Client_ID);
```

- **Employee Table Indexes**

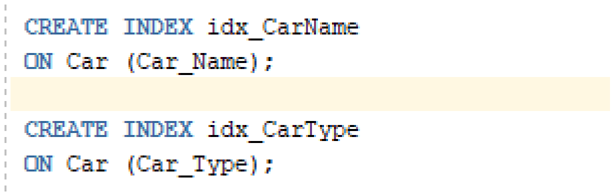
Indexes are placed on the Employee_ID and the subtype discriminator (Employee_Type). By this an employee will be identified easy and fast.



```
CREATE INDEX idx_EmployeeId
ON Employee (Employee_ID);

 CREATE INDEX idx_EmployeeType
ON Employee (Employee_Type);
```

- **Car Table Indexes**

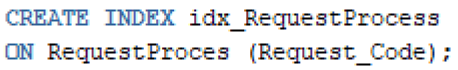
In this table, indexes are placed on Car_Name and Car_Type in order to speed up the retrieval of car information.



```
CREATE INDEX idx_CarName
ON Car (Car_Name);

CREATE INDEX idx_CarType
ON Car (Car_Type);
```

- **RequestProcess Table Indexes**

In this table, indexes will be placed on the Request_Code to show the progress of the request made by the receptionist.

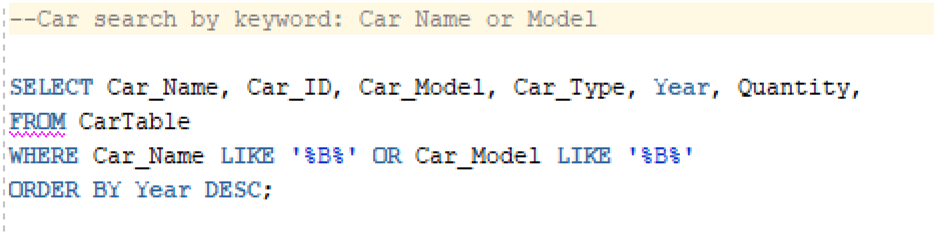


```
CREATE INDEX idx_RequestProcess
ON RequestProces (Request_Code);
```

SQL SELECT Statements and their Uses

Statement 1

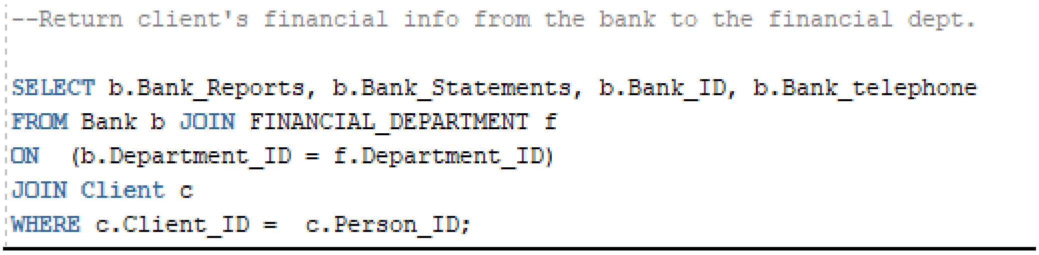
This statement is used to select the car name, car model, year and quantity from the car table. When the user (sales person) searches for the requested car. This allows the user to search words with keywords by the car name or car model.



```
--Car search by keyword: Car Name or Model

SELECT Car_Name, Car_ID, Car_Model, Car_Type, Year, Quantity,
FROM CarTable
WHERE Car_Name LIKE '%B%' OR Car_Model LIKE '%B%'
ORDER BY Year DESC;
```

Statement 2

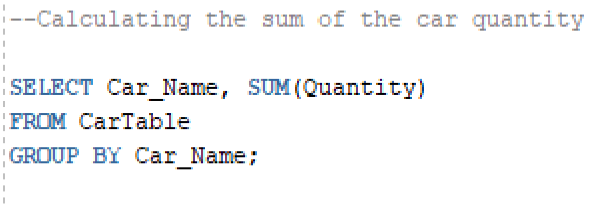
The statement allows to query information from the bank to the financial department. The statement selects the bank reports, bank statements, bank id and bank telephone. In order to ensure that information that is being sent corresponds with the correct user, the client id is checked against the person id and both comprise of the person's ID.



```
--Return client's financial info from the bank to the financial dept.

SELECT b.Bank_Reports, b.Bank_Statements, b.Bank_ID, b.Bank_telephone
FROM Bank b JOIN FINANCIAL_DEPARTMENT f
ON  (b.Department_ID = f.Department_ID)
JOIN Client c
WHERE c.Client_ID =  c.Person_ID;
```

Statement 3

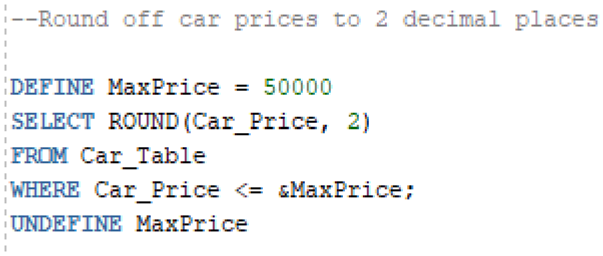
Since there are so many cars, the user would like to get the sum of the quantity of all cars with the same name. the sum of the quantity is grouped by the Car name.



```
--Calculating the sum of the car quantity

SELECT Car_Name, SUM(Quantity)
FROM CarTable
GROUP BY Car_Name;
```

Statement 4

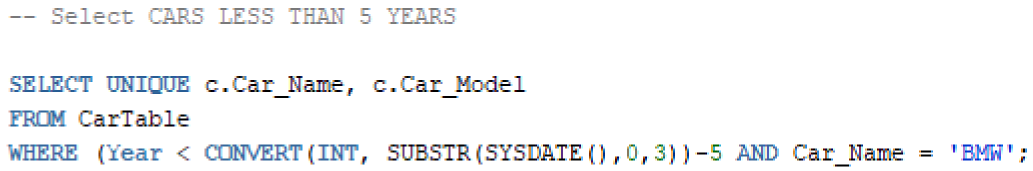
In order to ease up the searches of vehicles, the statement allows us to specify to search cars by max values. These searches will return vehicles with prices that are less or equal to the maximum value that was specified. All vehicle prices are rounded off to two decimal places for better readability.



```
--Round off car prices to 2 decimal places

DEFINE MaxPrice = 50000
SELECT ROUND(Car_Price, 2)
FROM Car_Table
WHERE Car_Price <= &MaxPrice;
UNDEFINE MaxPrice
```

Statement 5

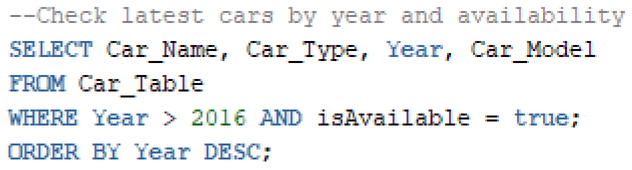
This statement allows us to select non-duplicate data of cars that are recent which are less than five years old. Here, the statement selects car name and car model. The statement also uses the system date function to get the current date and will do necessary calculation and get cars that were manufactured in less than five years. Furthermore, this statement allows the user to specify the car brand that they prefer. This means that the statement will query cars of a certain brand that was specified by the user but will only get recent cars (< 5 years).



```
-- Select CARS LESS THAN 5 YEARS

SELECT UNIQUE c.Car_Name, c.Car_Model
FROM CarTable
WHERE (Year < CONVERT(INT, SUBSTR(SYSDATE(),0,3))-5 AND Car_Name = 'BMW';
```

Statement 6

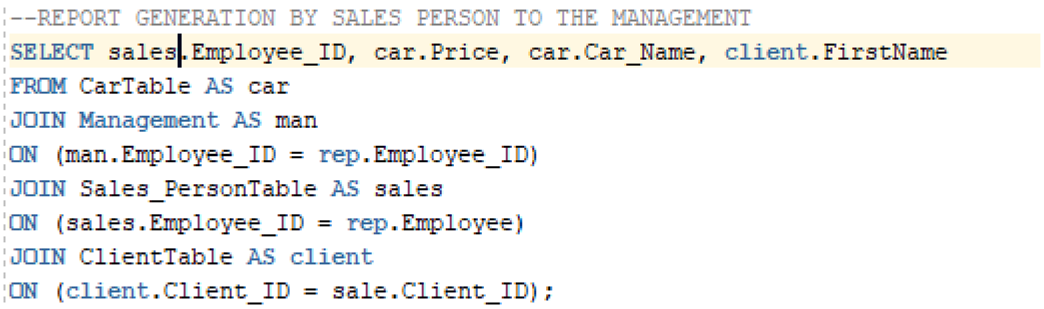
This statement allows us to select cars that were manufactured in 2016 or later. It selects the car name, car type, and car model from the car table. It only selects the cars that are currently available in the dealership. The ORDER BY query orders the cars in descending order by the years on which they were manufactured (i.e. From latest to oldest).



```
--Check latest cars by year and availability
SELECT Car_Name, Car_Type, Year, Car_Model
FROM Car_Table
WHERE Year > 2016 AND isAvailable = true;
ORDER BY Year DESC;
```

Statement 7

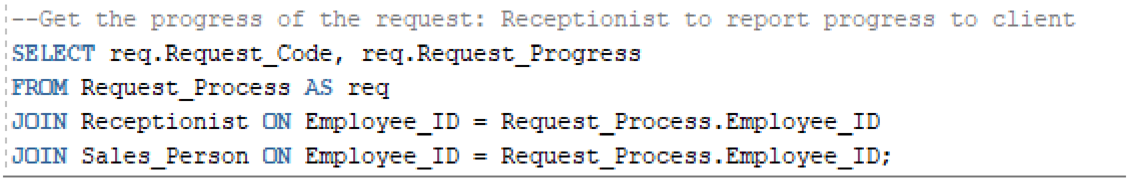
This statement allows a report to be generated by the sales person then sent to management. This report will display the following: employee id, car price, car name, date of report and the client’s first name. These attributes are taken from four different tables, namely: car table, management, sales person, and client table.



```
--REPORT GENERATION BY SALES PERSON TO THE MANAGEMENT
SELECT sales.Employee_ID, car.Price, car.Car_Name, client.FirstName
FROM CarTable AS car
JOIN Management AS man
ON (man.Employee_ID = rep.Employee_ID)
JOIN Sales_PersonTable AS sales
ON (sales.Employee_ID = rep.Employee)
JOIN ClientTable AS client
ON (client.Client_ID = sale.Client_ID);
```

Statement 8

This statement allows the receptionist to query the progress of the request that was made by the client. The statement selects only the request code and the request progress data of the request. This will allow the client to know how far the request is. The sales person will keep records of how the search of the car is going in the context of the specifications given by the client.



```
--Get the progress of the request: Receptionist to report progress to client
SELECT req.Request_Code, req.Request_Progress
FROM Request_Process AS req
JOIN Receptionist ON Employee_ID = Request_Process.Employee_ID
JOIN Sales_Person ON Employee_ID = Request_Process.Employee_ID;
```

Demonstration video's link: https://drive.google.com/file/d/116EM63ebGRPEcfpORMnXNVh6_LOL6LiK/view?usp=sharing.