**Logo

Description automatically generatedUniversity Of Bahrain**

**Department Of Computer Science**

**2021/2021 – First semester**

**ITCS496 – Physical implementation of DBMS**

**SHOWROOM DATABASE**

**MANAGEMENT SYSTEM**

**Table of Contents**

[**ER Diagram** 4](#_Toc90690329)

[**Relational Schema** 5](#_Toc90690330)

[**Creation of tables** 6](#_Toc90690331)

[**Insertion of values into table** 8](#_Toc90690332)

[**Tables** 10](#_Toc90690333)

[Customers 10](#_Toc90690334)

[Employee 10](#_Toc90690335)

[Showroom 10](#_Toc90690336)

[Cars 11](#_Toc90690337)

[Sale 11](#_Toc90690338)

[**Functions** 11](#_Toc90690339)

[Check if employee functions 11](#_Toc90690340)

[Check if car exists 13](#_Toc90690341)

[Check if a customer exits 14](#_Toc90690342)

[Check if showroom exists 15](#_Toc90690343)

[Check if employee is a manager 16](#_Toc90690344)

[Check if a car is sold 17](#_Toc90690345)

[Check if discount percentage is permitted 18](#_Toc90690346)

[Apply a discount 19](#_Toc90690347)

[**Procedures (and sequences)** 20](#_Toc90690348)

[Customer 20](#_Toc90690349)

[Customer sequence 20](#_Toc90690350)

[Procedure to add customer 20](#_Toc90690351)

[View for customer 21](#_Toc90690352)

[Procedure to view customer 22](#_Toc90690353)

[Procedure to update customer 23](#_Toc90690354)

[Procedure to delete a customer 24](#_Toc90690355)

[Employee Functions 24](#_Toc90690356)

[Sequence for employee 24](#_Toc90690357)

[Procedure to add a new employee 25](#_Toc90690358)

[Procedure to view employee 26](#_Toc90690359)

[Procedure to update employee 27](#_Toc90690360)

[Procedure to delete employee 28](#_Toc90690361)

[Showrooms 29](#_Toc90690362)

[Sequence for showroom 29](#_Toc90690363)

[Procedure to insert showroom 29](#_Toc90690364)

[Procedure to update showroom 30](#_Toc90690365)

[Procedure to view showroom 31](#_Toc90690366)

[Procedure to delete showroom 32](#_Toc90690367)

[Cars 33](#_Toc90690368)

[Sequence for car 33](#_Toc90690369)

[Procedure to insert car 33](#_Toc90690370)

[Procedure to view car 34](#_Toc90690371)

[Procedure to update car 35](#_Toc90690372)

[Procedure to delete car 36](#_Toc90690373)

[View to see only available cars (cars that are not sold) 37](#_Toc90690374)

[Sales 37](#_Toc90690375)

[Sales sequence 37](#_Toc90690376)

[Procedure to add a sale 38](#_Toc90690377)

[Procedure to view a sale 40](#_Toc90690378)

[**Form triggers** 42](#_Toc90690379)

[Check employee discount for manager 42](#_Toc90690380)

[Check employee discount for staff 43](#_Toc90690381)

[Trigger for car number 44](#_Toc90690382)

[Trigger to check showroom id 45](#_Toc90690383)

[Trigger to check if manager exists 46](#_Toc90690384)

[Trigger to insert employee 47](#_Toc90690385)

[Trigger to insert customer 48](#_Toc90690386)

[Trigger to insert car 49](#_Toc90690387)

[Trigger to insert sale 50](#_Toc90690388)

[**Login accounts** 50](#_Toc90690389)

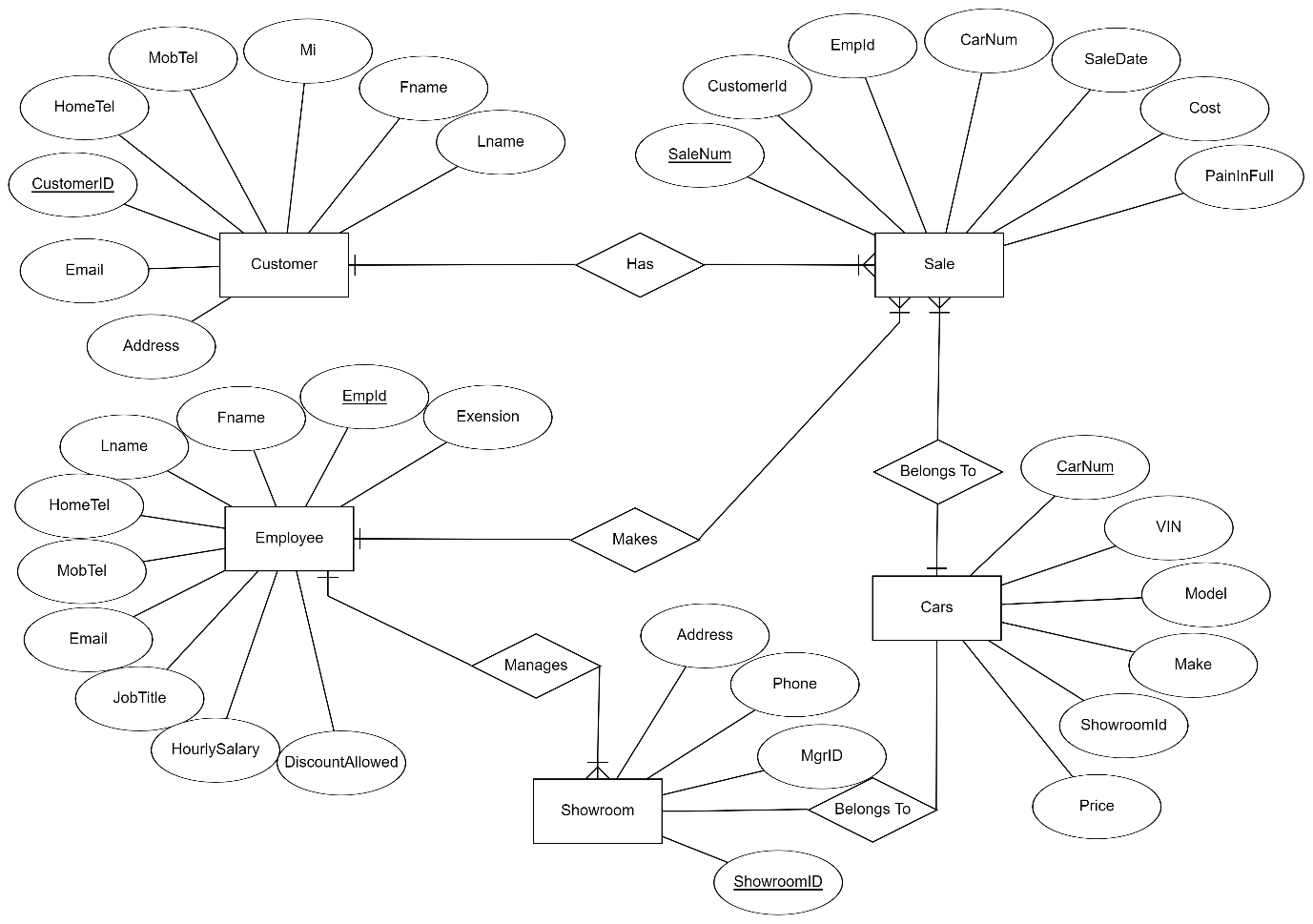
[**Application screenshots** 51](#_Toc90690390)

[Customer 51](#_Toc90690391)

[Staff 51](#_Toc90690392)

[Admin 54](#_Toc90690393)

# **ER Diagram**

****

**Assumptions**

* **Not all employees can make a sale.**
* **Not all employees can manage a showroom**
* **Not all customers have a sale.**

# **Relational Schema**

****

# **Creation of tables**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Script**

|  |
| --- |
| **DROP TABLE sale CASCADE CONSTRAINTS;**  **DROP TABLE cars CASCADE CONSTRAINTS;**  **DROP TABLE employee CASCADE CONSTRAINTS;**  **DROP TABLE customers CASCADE CONSTRAINTS;**  **DROP TABLE showroom CASCADE CONSTRAINTS;**  **--create table employee**  **CREATE TABLE employee**  **(**  **empId NUMBER(10)CONSTRAINTS employee\_empid\_pk PRIMARY KEY,**  **lastName VARCHAR2(15),**  **firstName VARCHAR2(15),**  **job\_title VARCHAR2(50),**  **email VARCHAR2(50),**  **mobTelNo VARCHAR2(10)CONSTRAINTS employee\_mobtelno\_nn NOT NULL,**  **HomeTel VARCHAR2(10),**  **extention VARCHAR2(32),**  **hourlySalary NUMBER(6,3),**  **discountAllowed NUMBER(3,2)**  **);**  **--create table customers**  **CREATE TABLE customers**  **(**  **customerID NUMBER(10)CONSTRAINTS customer\_customerid\_pk PRIMARY KEY,**  **lastName VARCHAR2(15),**  **firstName VARCHAR2(15),**  **mi VARCHAR2(10),**  **email VARCHAR2(50),**  **mobTel VARCHAR2(10)CONSTRAINTS customer\_mobtel\_nn NOT NULL,**  **homeTel VARCHAR2(10),**  **address VARCHAR2(30)CONSTRAINTS customer\_address\_nn NOT NULL**  **);**  **--create table showroom**  **CREATE TABLE showroom (**  **showroomID NUMBER(10)CONSTRAINTS showroom\_showroomid\_pk PRIMARY KEY,**  **MGRID NUMBER(10) CONSTRAINTS showroom\_mgrid\_fk REFERENCES employee(empId),**  **address VARCHAR2(30)CONSTRAINTS showroom\_address\_nn NOT NULL,**  **phone VARCHAR2(10)CONSTRAINTS showroom\_phone\_nn NOT NULL**  **);**  **--create table cars**  **CREATE TABLE cars**  **(**  **carNum NUMBER(10)CONSTRAINTS cars\_carnum\_pk PRIMARY KEY,**  **VIN VARCHAR2(50),**  **make DATE,**  **model VARCHAR2(20)CONSTRAINTS cars\_model\_nn NOT NULL,**  **Price NUMBER(8,3),**  **showroom\_id NUMBER(10) CONSTRAINTS cars\_showroom\_fk REFERENCES showroom (showroomID)**    **);**  **--create table sale**  **CREATE TABLE sale**  **(**  **saleNum NUMBER(10)CONSTRAINTS sale\_salenum\_pk PRIMARY KEY,**  **customerID NUMBER(10) CONSTRAINTS sale\_customerid\_fk REFERENCES customers (customerID),**  **empID NUMBER(10)CONSTRAINTS sale\_empid\_fk REFERENCES employee (empId),**  **carNum NUMBER(10)CONSTRAINTS sale\_carnum\_fk REFERENCES cars (carNum),**  **saleDate DATE,**  **cost NUMBER(8,3),**  **paidInfFull VARCHAR2(30)CONSTRAINTS paid\_nn NOT NULL,**  **additionalComments VARCHAR2(150)**  **);** |

# **Insertion of values into table**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Script**

|  |
| --- |
| **INSERT INTO customers**  **VALUES (1,'Shahjahan','Haris','a123','haris@gmail.com','39647363','17117364','Manama Road 3720 House 977');**  **INSERT INTO customers**  **VALUES (2,'Hamdan','Malik','a124','Hamdan@gmail.com','32210087','17742279','Manama Road 3720 House 977');**  **INSERT INTO customers**  **VALUES (3,'Haider','Syed','a125','syed@gmail.com','39110022','17227456','Sitra Road 0914 House 113');**  **INSERT INTO customers**  **VALUES (4,'Costa','Alex','a126','Alex@ymail.com','39653411','17635242','Budaiya Road 1010 Villa 120');**  **INSERT INTO customers**  **VALUES (5,'Talaq','Yousuf','a127','yousuf@gmail.com','39635344','17003735','Sitra Road 1002 Villa 110');**  **--create into table employee**  **INSERT INTO employee**  **VALUES(1,'Ali','Ahmed','staff','Hamood@hotmail.com' ,'39655432','17790052',NULL,20.000,0.03);**  **INSERT INTO employee**  **VALUES(2,'Ameen','Ameena','manager','AmeenaAmeen@gmail.com','39211934','17435546',NULL,60.000,0.07);**  **INSERT INTO employee**  **VALUES(3,'Sadiq','Mahmood','admin','mahmood.admin@gmail.com','32210097','17233456',NULL,90.000,0.07);**  **INSERT INTO employee**  **VALUES (4,'Hussain','Huda','staff','dodo\_bird@gmail.com','32110987','17112998',NULL,25.000,0.03);**  **INSERT INTO employee**  **VALUES (5,'Khalid','Isa','manager','IsaKhalid@gmail.com','32536647','17885034',NULL,50.000,0.07);**  **INSERT INTO showroom**  **VALUES(1,2,'Diplomatic Area','17520443');**  **INSERT INTO showroom**  **VALUES(2,5,'Riffa','17748839');**  **INSERT INTO cars**  **VALUES(1,'645372',TO\_DATE('2014','YYYY'),'Toyota Yaris',2500.500,1);**  **INSERT INTO cars**  **VALUES(2,'830637',TO\_DATE('2017','YYYY'),'Honda Civic',8000.300,1);**  **INSERT INTO cars**  **VALUES(3,'673632',TO\_DATE('2013','YYYY'),'KIA RIO',3500.500,2);**  **INSERT INTO cars**  **VALUES(4,'718172',TO\_DATE('2019','YYYY'),'Mazda 9',12500.500,2);**  **INSERT INTO cars**  **VALUES(5,'432424',TO\_DATE('2012','YYYY'),'Lexus LS350',16500.500,2);**  **INSERT INTO cars**  **VALUES(6,'424632',TO\_DATE('2014','YYYY'),'Toyota Camry',5500.500,2);**  **INSERT INTO cars**  **VALUES(7,'984324',TO\_DATE('2018','YYYY'),'Nissan Maxima',8500,1);**  **INSERT INTO cars**  **VALUES(8,'234523',TO\_DATE('2021','YYYY'),'Nissan Maxima',18500,1);**  **INSERT INTO cars**  **VALUES(9,'653567',TO\_DATE('2020','YYYY'),'Infiniy Q80',23500,1);**  **INSERT INTO cars**  **VALUES(10,'245242',TO\_DATE('2022','YYYY'),'Hyundai Sonata',8500,2);**  **INSERT INTO sale**  **VALUES(1,2,1,2,TO\_DATE('13-04-2018','DD-MM-YYYY'),8000.300,'NO',NULL);**  **INSERT INTO sale**  **VALUES(2,1,4,4,TO\_DATE('23-12-2019','DD-MM-YYYY'),12500.500,'NO',NULL);** |

# **Tables**

## Customers

**A screenshot of a computer

Description automatically generated with medium confidence**

## Employee

**A screenshot of a computer

Description automatically generated with medium confidence**

## Showroom

**Graphical user interface, application, Teams

Description automatically generated**

## Cars

**A screenshot of a computer

Description automatically generated**

## Sale

**Graphical user interface

Description automatically generated**

# **Functions**

## Check if employee functions

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE FUNCTION employee\_exists (id NUMBER )**  **RETURN BOOLEAN**  **IS**  **CURSOR emp\_cur IS SELECT \***  **FROM employee**  **WHERE empid = id;**  **e\_row emp\_cur%ROWTYPE;**  **BEGIN**  **OPEN emp\_cur;**  **Fetch emp\_cur INTO e\_row;**  **IF (emp\_cur%found) THEN**  **RETURN true;**  **Else**  **RETURN false;**  **END IF;**  **END;** |

## Check if car exists

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE FUNCTION car\_exists (id NUMBER )**  **RETURN BOOLEAN**  **IS**  **CURSOR car\_cur IS SELECT \***  **FROM cars**  **WHERE carnum = id;**  **c\_row car\_cur%ROWTYPE;**  **BEGIN**  **OPEN car\_cur;**  **Fetch car\_cur INTO c\_row;**  **IF (car\_cur%found) THEN**  **RETURN true;**  **Else**  **RETURN false;**  **END IF;**  **END;** |

## Check if a customer exits

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE FUNCTION customer\_exists (id NUMBER )**  **RETURN BOOLEAN**  **IS**  **CURSOR c\_cur IS SELECT \***  **FROM customers**  **WHERE customerid = id;**  **c\_row c\_cur%ROWTYPE;**  **BEGIN**  **OPEN c\_cur;**  **Fetch c\_cur INTO c\_row;**  **IF (c\_cur%found) THEN**  **RETURN true;**  **Else**  **RETURN false;**  **END IF;**  **END;** |

## Check if showroom exists

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE FUNCTION showroom\_exists (id NUMBER )**  **RETURN BOOLEAN**  **IS**  **CURSOR shr\_cur IS SELECT \***  **FROM showroom**  **WHERE showroomid = id;**  **sh\_row shr\_cur%ROWTYPE;**  **BEGIN**  **OPEN shr\_cur;**  **Fetch shr\_cur INTO sh\_row;**  **IF (shr\_cur%found) THEN**  **RETURN true;**  **Else**  **RETURN false;**  **END IF;**  **END;** |

## Check if employee is a manager

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE FUNCTION is\_manager (id NUMBER )**  **RETURN BOOLEAN**  **IS**  **CURSOR cur IS SELECT \***  **FROM employee**  **WHERE empid = id;**  **rec cur%ROWTYPE;**  **BEGIN**  **OPEN cur;**  **Fetch cur INTO rec;**  **IF (cur%found) THEN**  **IF UPPER(rec.job\_title) = 'MANAGER' THEN**  **RETURN true;**  **ELSE**  **RETURN false;**  **END IF;**  **Else**  **RETURN false;**  **END IF;**  **END;** |

## Check if a car is sold

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE FUNCTION car\_sold (id NUMBER )**  **RETURN BOOLEAN**  **IS**  **CURSOR sale\_cur IS SELECT \***  **FROM sale**  **WHERE carnum = id;**  **s\_row sale\_cur%ROWTYPE;**  **BEGIN**  **OPEN sale\_cur;**  **FETCH sale\_cur INTO s\_row;**  **IF (sale\_cur%found) THEN**  **RETURN true;**  **Else**  **RETURN false;**  **END IF;**  **END;** |

## Check if discount percentage is permitted

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE FUNCTION discount\_allowed (dis NUMBER, eid NUMBER )**  **RETURN BOOLEAN**  **IS**  **CURSOR cur IS SELECT \***  **FROM employee**  **WHERE empid = eid;**  **e\_row cur%ROWTYPE;**  **BEGIN**  **OPEN cur;**  **FETCH cur INTO e\_row;**  **IF e\_row.discountallowed >= dis AND dis >= 0 AND dis <= 1 THEN**  **RETURN true;**  **ELSE**  **RETURN false;**  **END IF;**  **END;** |

## Apply a discount

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE FUNCTION apply\_discount (dis NUMBER, price NUMBER )**  **RETURN NUMBER**  **IS**  **final NUMBER;**  **BEGIN**  **final := price - (price\*dis);**  **return final;**  **END;** |

# **Procedures (and sequences)**

## Customer

### Customer sequence

**A screenshot of a computer screen

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE SEQUENCE customer\_customerid\_seq**  **START WITH 6;** |

### Procedure to add customer

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE insert\_customer**  **(**  **lastname VARCHAR2,**  **firstname VARCHAR2,**  **mi VARCHAR2,**  **email VARCHAR2,**  **mobtel VARCHAR2,**  **hometel VARCHAR2,**  **address VARCHAR2**  **)**  **AS**  **BEGIN**  **INSERT INTO customers VALUES (customer\_customerid\_seq.nextval, lastname,**  **firstname, mi, email, mobtel, hometel, address);**  **COMMIT;**  **END;** |

### View for customer

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE VIEW view\_customer**  **AS SELECT customerid, lastname, firstname, mobtel, email FROM customers;** |

### Procedure to view customer

**A screenshot of a computer

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE view\_customer\_info( id NUMBER ) IS**  **CURSOR c1 IS**  **SELECT \* FROM view\_customer**  **WHERE customerid = id;**  **c\_rec c1%rowtype;**    **BEGIN**  **OPEN c1;**  **FETCH c1 INTO c\_rec;**    **IF c1%NOTFOUND THEN**  **DBMS\_OUTPUT.PUT\_LINE('No customer with this ID');**  **ELSE**  **DBMS\_OUTPUT.PUT\_LINE('Customer id: ' || c\_rec.customerid);**  **DBMS\_OUTPUT.PUT\_LINE('Customer name: ' || c\_rec.firstname || ' ' || c\_rec.lastname);**  **DBMS\_OUTPUT.PUT\_LINE('Mobile number: '||c\_rec.mobtel);**  **DBMS\_OUTPUT.PUT\_LINE('Email '||c\_rec.email);**  **END IF;**    **CLOSE c1;**  **END;** |

### Procedure to update customer

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE update\_customer**  **(**  **id NUMBER,**  **first VARCHAR2,**  **last VARCHAR2,**  **mobile VARCHAR2,**  **home VARCHAR2,**  **e VARCHAR2,**  **m VARCHAR2,**  **adr VARCHAR2**  **)**  **IS**  **BEGIN**  **UPDATE customers**  **SET firstname = first, lastname = last, mobtel = mobile,**  **hometel = home, email = e, mi = m, address = adr**  **WHERE customerid = id;**  **COMMIT;**  **END;** |

### Procedure to delete a customer

**Graphical user interface, text, application

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE delete\_customer (id NUMBER) IS**  **BEGIN**  **DELETE FROM customers**  **WHERE customerid=id;**  **COMMIT;**  **END;** |

## Employee Functions

### Sequence for employee

**A screenshot of a computer screen

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE SEQUENCE employee\_empid\_seq**  **START WITH 6;** |

### Procedure to add a new employee

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE insert\_employee**  **(**  **lastname VARCHAR2,**  **firstname VARCHAR2,**  **jt VARCHAR2,**  **email VARCHAR2,**  **mobtel VARCHAR2,**  **hometel VARCHAR2,**  **ext VARCHAR2,**  **salary NUMBER,**  **discount NUMBER**  **)**  **AS**  **BEGIN**  **INSERT INTO employee VALUES (employee\_empid\_seq.nextval, lastname,**  **firstname, jt, email, mobtel, hometel, ext, salary, discount);**  **COMMIT;**  **END;** |

### Procedure to view employee

**A screenshot of a computer

Description automatically generated with medium confidence**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE view\_employee( id NUMBER ) IS**  **CURSOR c1 IS**  **SELECT \* FROM employee**  **WHERE empid = id;**  **e\_rec c1%rowtype;**    **BEGIN**  **OPEN c1;**  **FETCH c1 INTO e\_rec;**    **IF c1%NOTFOUND THEN**  **DBMS\_OUTPUT.PUT\_LINE('No employee with this ID');**  **ELSE**  **DBMS\_OUTPUT.PUT\_LINE('Employee id: ' || e\_rec.empid);**  **DBMS\_OUTPUT.PUT\_LINE('Employee name: ' || e\_rec.firstname || ' ' || e\_rec.lastname);**  **DBMS\_OUTPUT.PUT\_LINE('Mobile number: '||e\_rec.mobtelno);**  **DBMS\_OUTPUT.PUT\_LINE('Email: '||e\_rec.email);**  **DBMS\_OUTPUT.PUT\_LINE('Hourly Salary: '||e\_rec.hourlysalary);**  **DBMS\_OUTPUT.PUT\_LINE('Job title: '||e\_rec.job\_title);**  **DBMS\_OUTPUT.PUT\_LINE('Discount allowed: '||e\_rec.discountallowed);**  **END IF;**    **CLOSE c1;**  **END;** |

### Procedure to update employee

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE update\_employee**  **(**  **id NUMBER,**  **last VARCHAR2,**  **first VARCHAR2,**  **jt VARCHAR2,**  **e VARCHAR2,**  **mobtel VARCHAR2,**  **home VARCHAR2,**  **ext VARCHAR2,**  **s NUMBER,**  **d NUMBER**  **)**  **IS**  **BEGIN**  **UPDATE employee**  **SET lastname = last, firstname = first, job\_title = jt, email = e, mobtelno = mobtel,**  **hometel = home, extention = ext, hourlysalary = s, discountallowed = d**  **WHERE empid = id;**  **COMMIT;**    **END;** |

### Procedure to delete employee

**Graphical user interface, text, application

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE delete\_employee (id NUMBER) IS**  **BEGIN**  **DELETE FROM employee**  **WHERE empid=id;**  **COMMIT;**  **END;** |

## Showrooms

### Sequence for showroom

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE SEQUENCE showroom\_showroomid\_seq**  **START WITH 3;** |

### Procedure to insert showroom

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE insert\_showroom**  **(**  **MGRID NUMBER,**  **address VARCHAR2,**  **phone NUMBER**  **)**  **AS**  **BEGIN**  **INSERT INTO showroom VALUES (showroom\_showroomid\_seq.nextval, MGRID, address,phone);**  **COMMIT;**  **END;** |

### Procedure to update showroom

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE update\_showroom**  **(**  **id NUMBER,**  **MID VARCHAR2,**  **adr VARCHAR2,**  **ph NUMBER**  **)**  **IS**  **BEGIN**  **UPDATE showroom**  **SET MGRID = MID, Address = adr, Phone = ph**  **WHERE showroomid = id;**  **COMMIT;**    **END;** |

### Procedure to view showroom

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE view\_showroom( id NUMBER ) IS**  **CURSOR c1 IS**  **SELECT \* FROM showroom**  **WHERE showroomid = id;**  **s\_rec c1%rowtype;**    **BEGIN**  **OPEN c1;**  **FETCH c1 INTO s\_rec;**    **IF c1%NOTFOUND THEN**  **DBMS\_OUTPUT.PUT\_LINE('No showroom with this ID');**  **ELSE**  **DBMS\_OUTPUT.PUT\_LINE('showroom id: ' || s\_rec.showroomid);**  **DBMS\_OUTPUT.PUT\_LINE('MGRID: ' || s\_rec.MGRID);**  **DBMS\_OUTPUT.PUT\_LINE('Address: '||s\_rec.address);**  **DBMS\_OUTPUT.PUT\_LINE('Phone '||s\_rec.phone);**  **END IF;**    **CLOSE c1;**  **END;** |

### Procedure to delete showroom

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE delete\_showroom (id NUMBER) IS**  **BEGIN**  **DELETE FROM showroom**  **WHERE showroomid=id;**  **COMMIT;**  **END;** |

## Cars

### Sequence for car

**A screenshot of a computer screen

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE SEQUENCE cars\_carnum\_seq**  **START WITH 11;** |

### Procedure to insert car

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **create or replace PROCEDURE insert\_car**  **(**  **num NUMBER,**  **v VARCHAR2,**  **m DATE,**  **model VARCHAR2,**  **p NUMBER,**  **shid NUMBER**  **)**  **AS**  **BEGIN**  **INSERT INTO cars VALUES (num, v, m, model, p, shid);**  **COMMIT;**  **END;** |

### Procedure to view car

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE view\_car( num NUMBER ) IS**  **CURSOR c1 IS**  **SELECT \* FROM cars**  **WHERE carnum = num;**  **rec c1%rowtype;**    **BEGIN**  **OPEN c1;**  **FETCH c1 INTO rec;**    **IF c1%NOTFOUND THEN**  **DBMS\_OUTPUT.PUT\_LINE('No car with this car number');**  **ELSE**  **DBMS\_OUTPUT.PUT\_LINE('Vehicle Idantification Number: ' || rec.vin);**  **DBMS\_OUTPUT.PUT\_LINE('Make: ' || rec.make);**  **DBMS\_OUTPUT.PUT\_LINE('Model: '||rec.model);**  **DBMS\_OUTPUT.PUT\_LINE('Show room: '||rec.showroom\_id);**  **END IF;**  **CLOSE c1;**  **END;** |

### Procedure to update car

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE update\_car**  **(**  **num NUMBER,**  **v VARCHAR2,**  **m DATE,**  **model VARCHAR2,**  **p NUMBER,**  **shid NUMBER**  **)**  **IS**  **BEGIN**  **UPDATE cars**  **SET vin = v, make = m, model = m, price = p,showroom\_id = shid**  **WHERE carnum = num;**  **COMMIT;**  **END;** |

### Procedure to delete car

**Graphical user interface, text, application

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE delete\_car (num NUMBER) IS**  **BEGIN**  **DELETE FROM cars**  **WHERE carnum=num;**  **COMMIT;**  **END;** |

### View to see only available cars (cars that are not sold)

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE VIEW available\_cars**  **AS SELECT \* FROM cars**  **WHERE carnum NOT IN (SELECT carnum FROM sale);**  **-----------------------**  **SELECT \* FROM available\_cars;** |

## Sales

### Sales sequence

**A screenshot of a computer screen

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE SEQUENCE salenum\_seq**  **START WITH 3;** |

### Procedure to add a sale

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE insert\_sale**  **(**  **cid NUMBER,**  **eid NUMBER,**  **cnum NUMBER,**  **paidinfull VARCHAR2,**  **cmt VARCHAR2,**  **dis VARCHAR**  **)**  **IS**  **price NUMBER := 0;**  **cost NUMBER := 0;**  **CURSOR c\_cur IS**  **SELECT price FROM cars**  **WHERE carnum = cnum;**    **rec c\_cur%rowtype;**  **BEGIN**    **IF NOT employee\_exists(eid) THEN**  **DBMS\_OUTPUT.PUT\_LINE('No employee with this ID.');**    **ELSIF NOT customer\_exists(cid) THEN**  **DBMS\_OUTPUT.PUT\_LINE('No customer with this ID.');**    **ELSIF NOT car\_exists(cnum) OR car\_sold(cnum) THEN**  **DBMS\_OUTPUT.PUT\_LINE('The car you entered is not available.');**    **ELSIF NOT discount\_allowed(dis, eid) THEN**  **DBMS\_OUTPUT.PUT\_LINE('You either entered an invalid discount amount, or the employee can''t make this much of a discount.');**  **DBMS\_OUTPUT.PUT\_LINE('Make sure the discount amount is between 0 and 1.');**  **ELSE**  **OPEN c\_cur;**  **FETCH c\_cur INTO rec;**  **price := rec.price;**    **IF dis = 0 THEN**  **cost := price;**  **ELSE**  **cost := apply\_discount (dis, price);**  **END IF;**    **INSERT INTO sale VALUES(salenum\_seq.nextval, cid, eid, cnum, sysdate, cost, paidinfull, cmt);**  **COMMIT;**    **DBMS\_OUTPUT.PUT\_LINE('Your sale was recorded successfully');**    **END IF;**  **END;** |

### Procedure to view a sale

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE PROCEDURE view\_sale( num NUMBER ) IS**  **CURSOR c1 IS**  **SELECT \* FROM sale**  **WHERE salenum = num;**  **rec c1%rowtype;**    **BEGIN**  **OPEN c1;**  **FETCH c1 INTO rec;**    **IF c1%NOTFOUND THEN**  **DBMS\_OUTPUT.PUT\_LINE('No sale with this ID');**  **ELSE**  **DBMS\_OUTPUT.PUT\_LINE('Sale number: ' || rec.salenum);**  **DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || rec.customerid);**  **DBMS\_OUTPUT.PUT\_LINE('Employee ID: '||rec.empid);**  **DBMS\_OUTPUT.PUT\_LINE('Car number: '||rec.carnum);**  **DBMS\_OUTPUT.PUT\_LINE('Sale date: '||rec.saledate);**  **DBMS\_OUTPUT.PUT\_LINE('Cost: '||rec.cost);**  **DBMS\_OUTPUT.PUT\_LINE('Paid in full: '||rec.paidinffull);**  **IF rec.additionalcomments IS NULL THEN**  **DBMS\_OUTPUT.PUT\_LINE('Additional comments: -');**  **ELSE**  **DBMS\_OUTPUT.PUT\_LINE('Additional comments: '||rec.additionalcomments);**  **END IF;**  **END IF;**  **CLOSE c1;**  **END;** |

# **Form triggers**

## Check employee discount for manager

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "EMPLOYEE\_DISC"**  **BEFORE**  **insert or update on "EMPLOYEE"**  **for each row**  **BEGIN**  **IF UPPER(:NEW.job\_title) = 'MANAGER' THEN**  **IF :NEW.discountallowed > 0.07 OR :NEW.discountallowed < 0 THEN**  **RAISE\_APPLICATION\_ERROR(-20001, 'Invalid manager discount amount.');**  **END IF;**  **END IF;**  **END;** |

## Check employee discount for staff

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "STAFF\_DISC"**  **BEFORE**  **insert or update on "EMPLOYEE"**  **for each row**  **BEGIN**  **IF UPPER(:NEW.job\_title) = 'STAFF' THEN**  **IF :NEW.discountallowed > 0.03 OR :NEW.discountallowed < 0 THEN**  **RAISE\_APPLICATION\_ERROR(-20002, 'Invalid staff discount amount. Should be less than 0.03%-');**  **END IF;**  **END IF;**  **END;** |

## Trigger for car number

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "CAR\_NUM"**  **BEFORE**  **insert on "CARS"**  **for each row**  **BEGIN**  **IF car\_exists(:NEW.carnum ) THEN**  **RAISE\_APPLICATION\_ERROR(-20003, 'A car with this car number already exists.');**  **END IF;**  **END;** |

## Trigger to check showroom id

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "CHECK\_SHOWROOM"**  **BEFORE**  **insert or update on "CARS"**  **for each row**  **BEGIN**  **IF NOT showroom\_exists(:NEW.carnum ) THEN**  **RAISE\_APPLICATION\_ERROR(-20004, 'Invalid showroom ID.');**  **END IF;**  **END;** |

## Trigger to check if manager exists

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "CHECK\_MANAGER"**  **BEFORE**  **insert or update on "SHOWROOM"**  **for each row**  **BEGIN**  **IF NOT is\_manager(:NEW.mgrid ) THEN**  **RAISE\_APPLICATION\_ERROR(-20006, 'No manager with this ID.');**  **END IF;**  **END;** |

## Trigger to insert employee

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "AUTO\_EMP"**  **BEFORE**  **insert or update on "EMPLOYEE"**  **for each row**  **BEGIN**  **IF :NEW.empid IS NULL THEN**  **SELECT employee\_empid\_seq.nextval**  **into :NEW.empid from sys.dual;**  **END IF;**  **END;** |

## Trigger to insert customer

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "AUTO\_CUST"**  **BEFORE**  **insert or update on "CUSTOMERS"**  **for each row**  **BEGIN**  **IF :NEW.CUSTOMERID IS NULL THEN**  **SELECT customer\_customerid\_seq.nextval**  **into :NEW.CUSTOMERID from sys.dual;**  **END IF;**  **END;** |

## Trigger to insert car

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "AUTO\_CAR"**  **BEFORE**  **insert or update on "CARS"**  **for each row**  **BEGIN**  **IF :NEW.CARNUM IS NULL THEN**  **SELECT cars\_carnum\_seq.nextval**  **into :NEW.CARNUM from sys.dual;**  **END IF;**  **END;** |

## Trigger to insert sale

**Text

Description automatically generated**

**Code**

|  |
| --- |
| **CREATE OR REPLACE EDITIONABLE TRIGGER "AUTO\_SALE"**  **BEFORE**  **insert or update on "SALE"**  **for each row**  **BEGIN**  **IF :NEW.SALENUM IS NULL THEN**  **SELECT salenum\_seq.nextval**  **into :NEW.SALENUM from sys.dual;**  **END IF;**  **END;** |

# **Login accounts**

1. **Customer**

|  |  |
| --- | --- |
| **Haris** | **123** |
| **Malik** | **123** |
| **Syed** | **123** |

1. **Staff**

|  |  |
| --- | --- |
| **Ahmed** | **123** |
| **Huda** | **123** |

1. **Manager**

|  |  |
| --- | --- |
| **Ameena** | **123** |
| **Isa** | **123** |

# **Application screenshots**

## Customer

1. **Home page**

****

1. **Edit account**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

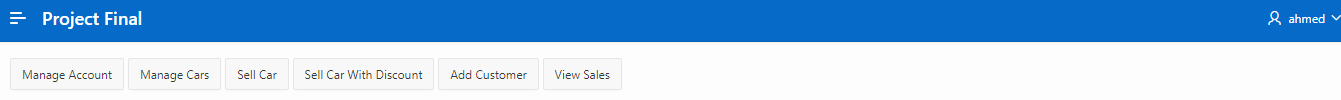
1. **View available cars**

**Graphical user interface, text, application

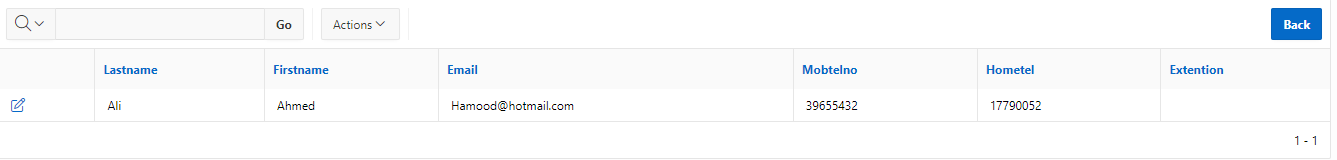
Description automatically generated**

## Staff

1. **Home page**

****

1. **Manage Account**

****

**Graphical user interface, application

Description automatically generated**

1. **Manage Cars**

**Graphical user interface, text, application

Description automatically generated**

1. **Sell car**

**Graphical user interface, application

Description automatically generated**

**A picture containing table

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface

Description automatically generated with low confidence**

1. **Sell car with discount**

**Graphical user interface, application

Description automatically generated**

1. **Add customer**

**Graphical user interface, application

Description automatically generated**

1. **View sales by customer (Top customer, bottom sale)**

**Graphical user interface, table

Description automatically generated**

## Admin

1. **Admin home**

****

1. **Manage Employees (View, Edit, Delete, Add)**

**A screenshot of a computer

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

1. **Manage Showrooms that you own (Create, Edit, Delete)**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application, Teams

Description automatically generated**

**Table

Description automatically generated with medium confidence**

1. **Manage Cars that you own (Create, Edit, Delete)**

**A screenshot of a computer

Description automatically generated**