

**School of Computer Science and Engineering** DBA Open ended experiment project Report On

**“Car showroom management’’**

|  |  |  |
| --- | --- | --- |
| USN | NAME | ROLL.NO |
| 01FE20BCS002 | Soumya K | 102 |
| 01FE20BCS006 | Neha Patil | 106 |
| 01FE20BCS007 | Lohit G | 107 |

**Content**

1. Introduction 3
   1. Car showroom management 3
   2. Database management system 4
   3. Problem statement 5
   4. Objectives 5
   5. Motivation 5
2. Requirement collection and analysis 6
   1. Introduction 6
   2. Data requirement 6
   3. Functional requirement 7
3. Database design 8
   1. Introduction 8
   2. Conceptual design 9
4. Implementation and results 12
   1. Introduction 12
   2. Results and references 15

**Chapter 1: Introduction**

**1.1 Car showroom management**

This Car showroom management system is designed for such owners and users who are interested in cars. This system keeps track of all the cars along with their each and every single detail. From the car models to up to what transactions can one negotiate while taking that car, one can view all the data here. If someone is interested in buying the car, then he/she can contact the retailer as well for the same, via the system.

Our project car showroom management system includes managing the details of cars, customers, employees and storing the details into the system in a computerize way. The modern industrial world is very much advanced in technology and the competition in the world is intense.

The management system has been developed to override the problems prevailing in the piratical manual system. This project is supported to eliminate and, in some case, reduce the hardships faced by the existing system. Moreover, this management system is designed for the particular need to carry out operations in a smooth and effective manner. This kind of system reduces as much as possible type of errors while inserting some data. Every organisation , whether big organisation or small organisation has challenged to overcome all the loop holes of its management system.

**1.2 Database management System**

Formally, a "database" refers to a set of related data and the way it is organised. Access to this data is usually provided by a "database management system" (DBMS) [3] consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data). The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organised.

DBMSs provide various functions that allow management of a database and its data which can be classified into

four main functional groups: • Data definition – Creation, modification and removal of definitions that define the

organization of the data.

• Update – Insertion, modification, and deletion of the actual data

* Retrieval – Providing information in a form directly usable or for further processing by other applications. The
* retrieved data may be made available in a form basically the same as it is stored in the database or in a new form
* obtained by altering or combining existing data from the database.
* Administration – Registering and monitoring users, enforcing data security, monitoring performance, maintaining
* data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event
* such as an unexpected system failure

**1.2.1 Relational database management**

Relational database managementsystem (RDBMS) RDBMS [3] is a database management system

* based on the relational model of data. Most databases in widespread use today are based on this model.
* RDBMSs have been a common option for the storage of information in databases used for financial records,
* manufacturing and logistical information, personnel data, and other applications since the 1980s. Relational
* databases have often replaced legacy data models like hierarchical databases and network databases because
* they were easier to implement and administer. Nonetheless, relational databases received continued,
* unsuccessful challenges by object database management systems in the 1980s and 1990s, (which were
* introduced in an attempt to address the so-called object-relational impedance mismatch between relational
* databases and object-oriented application programs), as well as by XML database management systems in
* the 1990s.However, due to the expanse of technologies, such as horizontal scaling of computer clusters,
* NoSQL databases have recently become popular as an alternative to RDBMS databases.

**1.3 PROBLEM STATEMENT**

* Consider a car showroom database where a sales manager manages transaction details, car models sold, and customer/car owner information. The following are the data requirements:
* SALES MANAGER ENTITY: Sales managers in a manufacturing organisation are uniquely identified by individual ids. In addition, the manager's name, phone number, address, age, and gender are also recorded.
* BRAND ENTITY: A manufacturing company with multiple sales managers is assigned a unique company id. In addition, the names of the brands (companies), the address where the company is situated, and the names of the proprietors(owner) too are recorded.
* TRANSACTION DETAILS ENTITY: Ids, which are unique identifiers for each transaction, are included in the transaction data. In addition, the transaction's description and details are all recorded.
* CAR MODEL ENTITY: The details of car models include unique car ids for each vehicle. The names of car models and car types are also recorded.
* CUSTOMER’S ENTITY: Sales consultants assist customers/car owners. Customer information, where each customer is uniquely identified by his/her own id, customer names, addresses, phone numbers, and email addresses are all recorded.

**1.4 Objectives of the project**

Following are the objectives of the project:

* To study and implement the basic database concepts.
* To understand the database design process.
* To explore the activities of Car showroom management for designing the database
* To study and familiar about the Structured Query Languages.

**1.5 Motivation**

The amount of data we produce every day is truly mind-boggling. There are about 2.5 quintillion bytes of data created each day at our current pace from the large organizations like Banking sector, educational sector, reservation sector, health care sector and many other business applications. Storing, Maintaining and using data for Decision making are the challenging issues. These issues motivate us to design and develop database application for storing and managing the daily activities of car management system.

**Chapter 2: Requirement Collection and Analysis**

**2.1 Introduction**

The most critical aspect of specification is the gathering and compilation of system and user requirements. This process is normally done in conjunction with managers and users. The major goal in requirements gathering process is to:

• Collect the data used by the organization,

• Identify relationships/conditions to be applied on the data,

• Identify future data needs, and

• Determine how the data is used and generated.

• Identify the functions that are performed on the data The starting place for data collection is gathering existing forms and reviewing policies and systems. Then, ask users what the data means, and determine their daily processes Following subsections discuss the data requirements and functional & nonfunctional requirements identified based on the following activities collected from the car show room management users.

• The car showroom management has multiple branches in around the city, each branch is uniquely identified by its name, ID and a particular location in the city.

• The show room keeps name, date of birth, address of every customer. Every customer is assigned with unique number.

• show room also keeps every employee’s name, address, gender, date of birth. Every employee is uniquely identified by the employee number.

• Show room maintains many cars, each car has Registration number, Model, type.

• Employee can work for only one branch but a branch may have multiple employees. Showroom also keeps the record of transaction details which contains time, date, id, company id, car id, customer id.

**2.2 Data requirement**

Data requirement describes the data to be stored in the database pertaining to activities of the car showroom management requirements.

|  |  |  |
| --- | --- | --- |
| Sl.No | Entities | Attributes |
| 1 | Customers | Id, name, address, phone, email |
| 2 | Transaction | Date, time, id, tid, Cid , carid, sid , cuid |
| 3 | Company | Name, id, address, phone |
| 4 | Car model | Id, name, model name, |
| 5 | Store | Name, address, phone |
| 6 | Sales consultant | Name, id, phone, address, gender, age, cid |

**2.3 Functional requirements**

Functional requirementsare product features or functions that developers must implement to enable users to accomplish their tasks. So, it’s important to make them clear both for the development team and the stakeholders (clients). Table 2.3 shows the different types users of driving school database application and their respective responsibilities (tasks). Table 2.4 shows the different functions and user can perform on the database

|  |  |  |
| --- | --- | --- |
| Sl.no | Users | Responsibilities |
| 1 | Customer | Responsible for Viewing and modifying every branch data |
| 2 | Sales consultant | Responsible for Viewing and modifying every branch data |
| 3 | Administrator | Data administration functionalities |
| 4 | Branch Assistants | Responsible for enrolment of customer, adding employees, adding cars and generating various reports as per the requirement of Branch/School head/customer/employees etc. |

**Chapter 3: Database design**

**3.1 Introduction**

The requirements gathering and specification provides you with a high-level understanding of the organization, its data, and the processes that you must model in the database. Database design involves constructing a suitable model of for the information. Since the design process is complicated, especially for large databases, database design is divided into three phases:

• Conceptual database design

• Logical database design

• Physical database design.

In our project work we are addressing the conceptual database design using ER modeling and logical database design using the implementation data model called Relational model.

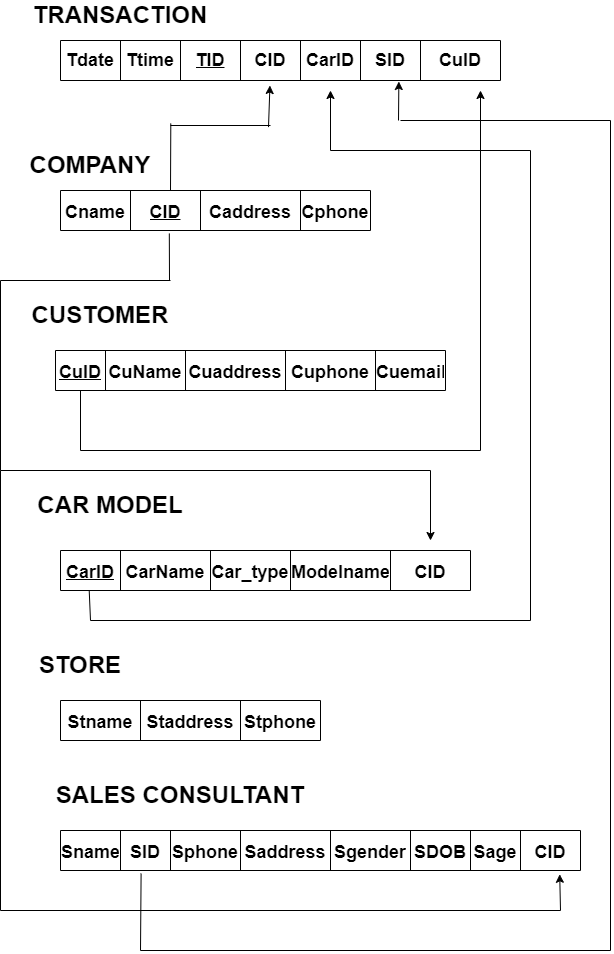
**3.2 Conceptual Database Design**

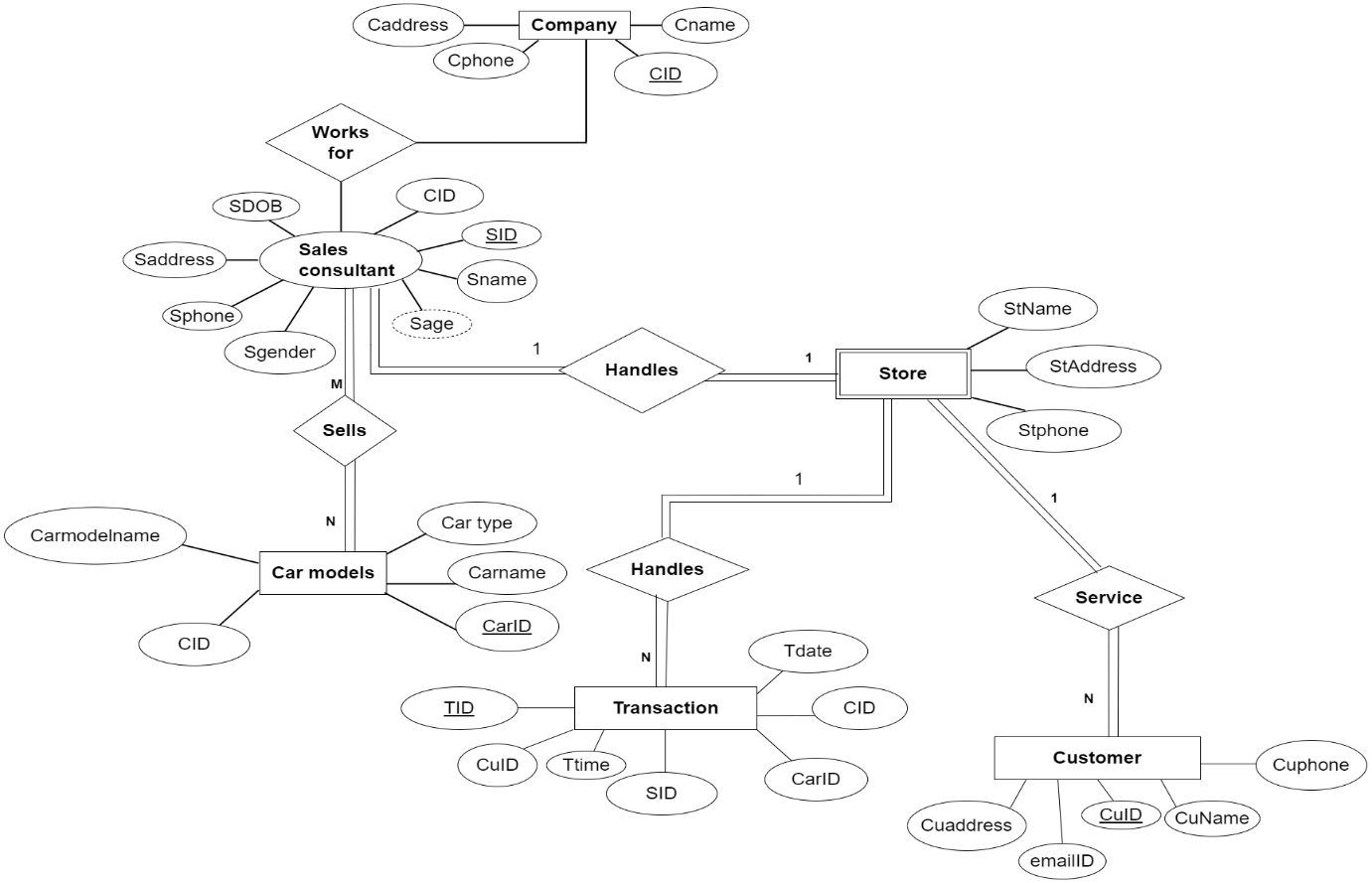
Conceptual database design involves modelling the collected information at a high-level of abstraction without using a particular data model or DBMS. This model allows for easy communication between end-users and database developers and has a clear method to convert from high-level model to relational model. The most popular model for conceptual database design is the Entity Relationship model which describes data as attribute, entity and relationship.

**List of entity types**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.no | Entity type name | Type of entity type | Justification |
| 1 | Transaction | Strong entity | It has a key attribute |
| 2 | Company | Strong entity | It has a key attribute |
| 3 | Customer | Strong entity | It has a key attribute |
| 4 | Car model | Strong entity | It has a key attribute |
| 5 | Store | Weak entity | It does not have a key attribute |
| 6 | Sales consultant | Strong entity | It has a key attribute |

**3.2.1 Relational schema, ER diagram and normalization**





CUSTOMER (cuid, cuname,cuaddress, cuphone , cuemail)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

TRANSACTION (tdate,ttime,tid , cid , carid , sid, cuid)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

COMPANY (cid, cname,caddress, cphone )

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

CAR MODEL (carid, carname,carmodelname, cid)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

STORE (stname , staddress, stphone)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

SALES CONSULTANT (sname,sid, sphone , saddress , sgender , sdob , sage ,cid)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

**Chapter 4: Implementation and Results**

**4.1 Introduction**

Implementation involves the construction of a database according to the specification of a logical schema. This will include the specification of an appropriate storage schema, security enforcement, external schema and so on. Implementation is influenced by the choice of available DBMSs, database tools and operating environment. There are additional tasks beyond simply creating a database schema and implementing the constraints such as data must be entered into the tables, issues relating to the users and user processes need to be addressed, and the management activities associated with wider aspects of corporate data management need to be supported. In practice, implementation of the logical schema in a given DBMS requires a very detailed knowledge of the specific features and facilities that the DBMS has to offer. In an ideal world, and in keeping with good software engineering practice, the first stage of implementation would involve matching the design requirements with the best available implementing tools and then using those tools for the implementation. In database terms, this might involve choosing vendor products with DBMS and SQL variants most suited to the database which is to be implemented. There are many relational DBMSs, available such as Oracle Database, Microsoft SQL Server, MySQL, IBM DB2, IBM Informix and Microsoft Access, use SQL. In this project we used Oracle SQL developer create the following tables of car showroom management.

Creation of database:

drop table company cascade constraint;

drop table customer cascade constraint;

drop table carmodel cascade constraint;

drop table salesconsultant cascade constraint;

drop table transactions cascade constraint;

drop table stores cascade constraint;

create table company

(

cid int primary key,

cname varchar(20) unique,

caddress varchar(50),

cphone int

);

create table customer

(

cuid int primary key,

cuname varchar(20) unique,

cuaddress varchar(50),

cuphone int,

cuemail varchar(30)

);

create table carmodel

(

carid int primary key,

carname varchar(30),

cartype varchar(30),

carmodelname varchar(30),

cid int references company(cid)

);

create table salesconsultant

(

sname varchar(20),

scid int primary key,

sphone int,

saddress varchar(50),

sgender varchar(10),

sdob date,

sage int,

cid int references company(cid)

);

create table transactions

(

tid int primary key,

tdate date,

ttime int,

cid int references company(cid),

carid int references carmodel(carid),

scid int references salesconsultant(scid),

cuid int references customer(cuid)

);

create table stores

(

stname varchar(10),

staddress varchar(50),

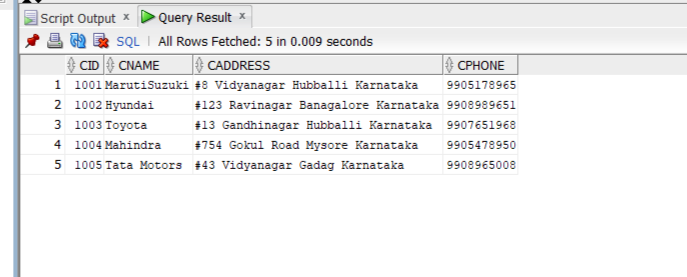
stphone int,

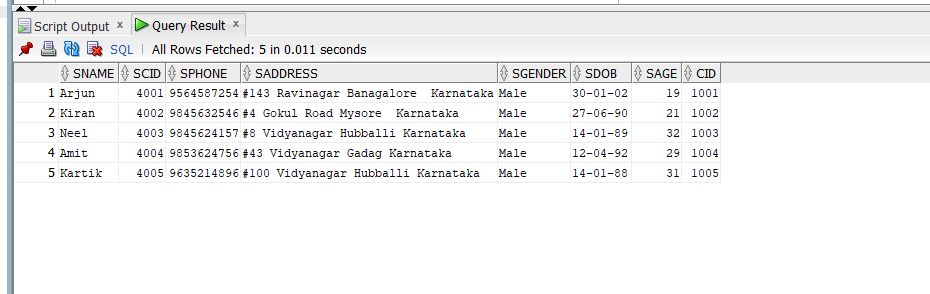
stid int primary key,

cid int references company(cid)

);

**4.2 Result**





**References:**

1. <https://www.freeprojectz.com/paid-projects/php-mysql/car-showroom-management-system-mini-project>
2. <https://phpgurukul.com/car-showroom-management-system-using-php-and-mysql/>
3. <https://www.scribd.com/document/337853350/car-Showroom-Management-System-doc>

**THANK YOU**