# INDEX

1. An answer to the following question, together with a justification for the decision: In its current form, it is a traditional database. Should you keep it that way?
2. The detailed process of implementing the database using SQL, including the normalisation of the table (should that be required), the identification of the attributes, the Entity-Relationship Diagram, and the use of SQL commands in order to create and populate the tables with data.
3. The SQL code for the following queries together with an explanation of the code:

1. Find the supervisors of the branches which rented out SUVs – output the supervisor’s ID and name.

2. Find the supervisors of the branches which had bills higher than 500 – output the supervisor’s ID and name.

3. Find the supervisors of the branches which rented out cars in May 2021 – output the supervisor’s ID and name.

4. Find the supervisors of the branches which handed out penalties – output the supervisor’s ID and name.

5. Find the type of car that was rented the highest number of times – output the car type and the number of times.

D. A choice of 4 types of graphs suitable for analysing the information in the database, their Implementation in Python (both the code and the output graph itself should be included in the report) and an explanation of what the graphs illustrate in your own words

A=Normalization is a technique for organizing data in a database. It is important that a database is normalized as it minimizes redundancy and to ensure only related data is stored in each table. Also, It helps as Updates run quickly due to no data being duplicated in multiple locations, inserting data run quickly since there is only a single insertion point for a piece of data and no duplication is required, Tables are typically smaller than the tables found in non-normalized databases. This usually allows the tables to fit into the buffer, thus offering faster performance. As compared to unnormalized database there is more redundant data therefore updates are slower when inserting data u do not need to worry about duplicate data and worse performance overall.

B=

**Table

Description automatically generated**

**1nf: Graphical user interface, application, table, Excel

Description automatically generated**

**Bill nr --PK and FK bill nr -- PK**

**2nf: kGraphical user interface, application

Description automatically generated**

**Bill nr --PK and FK branch code --PK bill nr-- PK**

**Branch code --FK**

**3nf:**

**Bill nr -- PK and FK bill nr, branch code and car late nr-- PK car plate nr --PK**

**Bill nr-- FK**

Table

Description automatically generated Table

Description automatically generatedTable

Description automatically generated

Bill detail linker table car detail

Table

Description automatically generated

Branch info

**Branch code -- PK**

Fist of all I split table in two with bill nr as primary key then I took branch name ,banch id ,supervsior id ,superviosr name in seprate table with branch code as priamry key , then linked bill nr and branch code and put them in one table and lastly I seprtaed car plate nr(PK) ,car description and car stype in seprate tabel and put bill nr(PK) brancah code and car plate nr togather in table so everthing links up with each other.

Then I created a database through SQLite studio 3 and used python to create table and insert data into table

        c.execute("""CREATE TABLE IF NOT EXISTS "branch\_info" ("branch\_code" INTEGER,"branch\_name" CHAR,"supervisor\_id"INTEGER,"supervisor\_name" CHAR,PRIMARY KEY("branch\_code"))""")

        conect.commit()

        c.execute("""INSERT INTO branch\_info ( branch\_code, branch\_name, supervisor\_id, supervisor\_name )

                    VALUES

                    (100023,'Coventry',871,'Anna Smith'),

                    (456109,'Leamington Spa',149,'John Cruise'),

                    (555901,'Wolverhampton',111,'Catherine Johnson'),

                    (876734,'Walsall',102,'David Brown'),

                    (981256,'Warwick',823,'James Doherty')""")

        conect.commit()

Firstly, I used I created table using CARETE table SQL command and named table branch info and then made column such as branch code (integer) which is PRNAMRY KEY ,branch name (char) ,supervisor id (integer), supervisor name (char) and then I committed the changes. After that I inserted data in table using INSERT command and committed changes.

        c.execute("CREATE TABLE IF NOT EXISTS car\_detail (car\_plate\_nr VARCHAR, car\_description VARCHAR, car\_type CHAR, PRIMARY KEY (car\_plate\_nr))")

        conect.commit()

        c.execute("""INSERT INTO car\_detail ( car\_plate\_nr, car\_description, car\_type)

                VALUES

                ('DS4049','BA1234','SUV'),

                ('DL3434','BA6753','Sports\_Car'),

                ('OP9817','BA1561','SUV'),

                ('SJ7182','BA9878','Hatchback'),

                ('BN9745','BA9123','SUV'),

                ('LA5142','BA8177','Sedan'),

                ('CB0098','BA4545','Sports\_Car'),

                ('ZX7222','BA1000','Coupe'),

                ('QW0128','BA8882','Sedan'),

                ('PO8123','BA5656','SUV'),

                ('IU7878','BA0012','Hatchback'),

                ('GF5612','BA3421','Sedan'),

                ('NM8787','BA4545','Sports\_Car'),

                ('VC1111','BA8177','Sedan'),

                ('FG7100','BA9123','Hatchback'),

                ('RE6000','BA9878','Sedan'),

                ('TR6199','BA1561','SUV'),

                ('DR1166','BA6753','Sports\_Car'),

                ('BP9111','BA1234','Coupe')

                """)

        conect.commit()

Similarly, to before I used create table SQL command to create command with column car plate nr (varchar ) which Is PK , car description (varchar) and car type (char) ,committed changes and inserted data using INSERT command and committed changes.

conect=sqlite3.connect("C:/Users/Talha/Desktop/R/4005/test.db")

        c=conect.cursor()

        c.execute("""CREATE TABLE IF NOT EXISTS bill\_detail (bill\_nr INTEGER, bill\_date DATE, penalty INTEGER, final\_bill INTEGER, PRIMARY KEY (bill\_nr))""")

        conect.commit()

        c.execute("""INSERT INTO bill\_detail ( bill\_nr, bill\_date,penalty, final\_bill)

                VALUES

                (166651, '18/01/2021',50,1050),

                (123111, '19/02/2021',0,500),

                (561909, '06/03/2021',0,480),

                (565690, '29/01/2021',0,680),

                (128976, '10/10/2021',0,710),

                (511899, '25/11/2021',20,1500),

                (141421, '03/12/2021',0,850),

                (514879, '29/10/2021',0,1250),

                (771100, '16/11/2021',20,300),

                (675912, '06/01/2022',50,350),

                (991762, '08/02/2022',0,950),

                (110054, '19/07/2021',100,1400),

                (378123, '12/08/2021',20,450),

                (808051, '18/09/2021',0,670),

                (100023, '21/07/2021',0,1030),

                (611554, '27/08/2021',50,520),

                (888712, '10/04/2021',0,490),

                (343412, '28/05/2021',20,1230),

                (222678, '04/06/2021',0,1680)

                """)

        conect.commit()

Similarly, to before I used create table SQL command to create command with column bill nr(integer) which Is PK , car description (varchar) and bill date (DATE), penalty (integer) , final bill (integer ) ,committed changes and inserted data using INSERT command and committed changes.

       c.execute("""CREATE TABLE IF NOT EXISTS linker\_table (bill\_nr INTEGER PRIMARY KEY REFERENCES bill\_detail (bill\_nr) MATCH "FULL", branch\_code INTEGER REFERENCES branch\_info (branch\_code) MATCH SIMPLE, car\_plate\_nr VARCHAR REFERENCES car\_detail (car\_plate\_nr) MATCH "FULL")""")

        conect.commit()

        c.execute("""INSERT INTO linker\_table  ( bill\_nr, branch\_code, car\_plate\_nr)

                VALUES

                (166651,876734, 'DS4049'),

                (123111,876734, 'DL3434'),

                (561909,876734, 'OP9817'),

                (565690,876734, 'SJ7182'),

                (128976,100023, 'BN9745'),

                (511899,100023, 'LA5142'),

                (141421,100023, 'CB0098'),

                (514879,100023, 'ZX7222'),

                (771100,100023, 'QW0128'),

                (675912,456109, 'PO8123'),

                (991762,456109, 'IU7878'),

                (110054,981256, 'GF5612'),

                (378123,981256, 'NM8787'),

                (808051,981256, 'VC1111'),

                (100023,981256, 'FG7100'),

                (611554,981256, 'RE6000'),

                (888712,555901, 'TR6199'),

                (343412,555901, 'DR1166'),

                (222678,555901, 'BP9111')

                """)

        conect.commit()

lastly, like before I used create table SQL command to create command with column bill nr(integer) which Is PK and FK, branch code (integer) Fk and car plate nr (varchar) ,committed changes and inserted data using INSERT command , with REFERENCES command to link with primary key of other table and committed changes.

C.1=

    conect=sqlite3.connect("C:/Users/Talha/Desktop/R/4005/test.db")

    c=conect.cursor()

    suvs=c.execute("""SELECT car\_plate\_nr FROM car\_detail WHERE car\_type ="SUV" """).fetchall()

    list\_suv=tuple(i[0] for i in suvs)

    lengthofsuvlist=len(list\_suv)

    print("  Id    Name")

    i=0

    while i < lengthofsuvlist:

        branchcode=c.execute("""SELECT branch\_code FROM linker\_table WHERE car\_plate\_nr =? """,(list\_suv[i],)).fetchall()

        listbranchcode=tuple(i[0] for i in branchcode)

        superviosrname=c.execute("""SELECT supervisor\_name  FROM branch\_info WHERE branch\_code =? """,(listbranchcode[0],)).fetchall()

        superviosrid=c.execute("""SELECT supervisor\_id  FROM branch\_info WHERE branch\_code =? """,(listbranchcode[0],)).fetchall()

        print(superviosrid[0]+superviosrname[0])

        i=i+1

    conect.commit()

    conect.close()

C.2=

    conect=sqlite3.connect("C:/Users/Talha/Desktop/R/4005/test.db")

    c=conect.cursor()

    bill=c.execute("""SELECT bill\_nr FROM bill\_detail WHERE final\_bill >500 """).fetchall()

    list\_bill=tuple(i[0] for i in bill)

    lengthbillnr=len(list\_bill)

    print("  Id    Name")

    i=0

    while i < lengthbillnr:

        branchcode=c.execute("""SELECT branch\_code FROM linker\_table WHERE bill\_nr =? """,(list\_bill[i],)).fetchall()

        listbranchcode=tuple(i[0] for i in branchcode)

        superviosrname=c.execute("""SELECT supervisor\_name FROM branch\_info WHERE branch\_code =? """,(listbranchcode[0],)).fetchall()

        superviosrid=c.execute("""SELECT supervisor\_id  FROM branch\_info WHERE branch\_code =? """,(listbranchcode[0],)).fetchall()

        print(superviosrid[0]+superviosrname[0])

        i=i+1

    conect.commit()

    conect.close()

C.3=

    conect=sqlite3.connect("C:/Users/Talha/Desktop/R/4005/test.db")

    c=conect.cursor()

    billnr=c.execute("""SELECT bill\_nr FROM bill\_detail WHERE bill\_date >='01/05/2021' AND bill\_date <= '31/05/2021' """).fetchall()

    list\_billnr=tuple(i[0] for i in billnr)

    lengthbillnr=len(list\_billnr)

    print("  Id    Name")

    i=0

    while i < lengthbillnr:

        branchcode=c.execute("""SELECT branch\_code FROM linker\_table WHERE bill\_nr =? """,(list\_billnr[i],)).fetchall()

        listbranchcode=tuple(i[0] for i in branchcode)

        superviosrname=c.execute("""SELECT supervisor\_name FROM branch\_info WHERE branch\_code =? """,(listbranchcode[0],)).fetchall()

        superviosrid=c.execute("""SELECT supervisor\_id  FROM branch\_info WHERE branch\_code =? """,(listbranchcode[0],)).fetchall()

        print(superviosrid[0]+superviosrname[0])

        i=i+1

    conect.commit()

    conect.close()

C.4=

    conect=sqlite3.connect("C:/Users/Talha/Desktop/R/4005/test.db")

    c=conect.cursor()

    billpenalty=c.execute("""SELECT bill\_nr FROM bill\_detail WHERE penalty > 0 """).fetchall()

    list\_billpenalty=tuple(i[0] for i in billpenalty)

    lengthbillpenalty=len(list\_billpenalty)

    print("  Id    Name")

    i=0

    while i < lengthbillpenalty:

        branchcode=c.execute("""SELECT branch\_code FROM linker\_table WHERE bill\_nr =? """,(list\_billpenalty[i],)).fetchall()

        listbranchcode=tuple(i[0] for i in branchcode)

        superviosrname=c.execute("""SELECT supervisor\_name FROM branch\_info WHERE branch\_code =? """,(listbranchcode[0],)).fetchall()

        superviosrid=c.execute("""SELECT supervisor\_id  FROM branch\_info WHERE branch\_code =? """,(listbranchcode[0],)).fetchall()

        print(superviosrid[0]+superviosrname[0])

        i=i+1

    conect.commit()

    conect.close()

C.5=

conect=sqlite3.connect("C:/Users/Talha/Desktop/R/4005/test.db")

    c=conect.cursor()

    mostrented=c.execute("""SELECT car\_type FROM car\_detail """).fetchall()

    list\_mostrented=tuple(i[0] for i in mostrented)

    c=Counter(list\_mostrented)

    print(c)