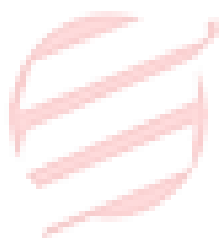


# Database normalizationn



EssayCorp **5** years  
★★★★★

## Question 1

Here the users of the DBMS are customers. The Customers data needs to be stored as a customer table. The customer table will be having all the details about the customer like customer number, customer name, customer phone number and customer address. As given in the scenario Customer address may not be present. The value which is very unique should be the primary key. As per that definition, each and every customer will be having customer number. Not two customers will be having same customer number and not two customer numbers can go to a same customer. Hence customer number will be considered for a primary key. The customer number can have customer address along with phone number. Most of the data fields are variables and they are CHAR type. Only the phone number can be a number and it is an INTEGER type.

1(b) (i)

The preceding steps contain the database design process:

- 1. Firstly, determine your database's purpose**

It helps you to prepare for the rest of the steps.

- 2. Secondly search and arrange the required information**

Collect all the types of information you would require for recording in the database like name of the product and its order number.

- 3. Then create table and segregate all the information**

Information will be divided into entities or subjects. Each entity will result a table

- 4. Next create columns for the information items**

Choose what data you need to store in every table. Everything turns into a field, and is shown as a section in the table. For instance, an Employees table may incorporate fields, for example, Last Name and Hire Date.

- 5. Further mention the primary keys**

Pick every table's primary key which is a segment that is utilized to interestingly distinguish every column. A case may be Product ID or Order ID.

### **6. Then next set up the table relationships**

Take a gander at every table and choose how the information in one table is identified with the information in different tables. Add fields to tables or make new tables to illuminate the connections, as important.

### **7. And then refine your design**

Investigate your outline for mistakes. Make the tables and include a couple records of test information. Check whether you can get the outcomes you need from your tables. Do the adjustments as per the requirement.

### **8. And apply the normalization rules**

Normalization rules needs to be applied and the results needs to be obtained.

1(b) (ii)

Almost any database needs a client interface to include, recover, overhaul and erase records, known by the ugly acronym CRUD. There are various administrator applications that give such nonexclusive facilities. For instance, for PostgreSQL we have pgAdmin which offers a GUI and phpPgAdmin and Adminer which offer PHP web interfaces. These have a few impediments.

The improvement of user interfaces (UIs), extending from ahead of schedule prerequisites to programming out of date quality, has turned into a period expending and excessive procedure. Normally, the graphical user interface (GUI) of an intelligent framework speaks for around 48% of the source code, requires around 45% of the advancement time and half of the usage time and covers 37% of the support time. These figures, assessed in the mid-nineties, are expanding significantly with the spread of new communication systems, for example, vocal and gestural modalities, bringing about extra necessities.

### **Distributed Database Concept**

CPU is a memory which has database that is detached to handling unit. It screens by the dispersed framework of database administration. The information is stored in different PCs. They might vanish at an end of systems of ward PCs and might likewise stay in the first areas. In conflicting to the parallel frameworks that has firmly coupled processor and build up

single database framework then the circulated database framework are approximately combined without segments actually associated physically.

In the distributed database the information is orchestrated with the assistance of framework managers. Since information are put away everywhere throughout the systems the information is available broadly everywhere throughout the systems such as system servers, on other organization systems, extranets and the corporate intranets. The appropriated database permits just the customer to execute the dates' and to handle particular data's.

Two sorts of procedures are there to set up distributed database that are close to date and constant, they are: 1. Replication and 2. Duplication

**Replication:**

In the distributive database the reasonable programming ventures gets copied and hunt down any modification in the database. This system keeps the entire database same and it is a compelling strategy in which it ingests extra time in light of the range of the database in the appropriated structure. It depends on the amount of databases arranged in the structure. From now on to do this action we must need package of time and resources.

**Duplication:**

In the distributive framework this instrument recognizes a database and equivalents that database. It sets time for duplication of information's and to acclimate that the dispersed territory has the first information or not. The client in database just adjusts the expert database and thus it will acclimate about the nearby information which is not adjusted. These two systems provide the information consistently in each conveyed area.

## Question 2

Database is nothing collection table about an Entity. Table is an intersection of rows and columns or it can also be defined as collection of interrelated Records. Record is nothing collection related filed about an entity for example details of a person or book etc.

First identify related tables to design a database. For example, here we can create Employee table with fields

Employee

S.No	Attribute	Datatype	Integrity Constraint
1	Emp_No	Number(6)	Primary Key
2	Name	Varchar2(20)	Not Null
3	Telephone	Number(10)	
4	Category	Varchar2(5)	Check Constraint

We can use DDL (Data Definition Language) and do lot of tasks. This DDL can be used for database creation, alteration and objects removal. CREATE, DROP, ALTER are the most commonly used DDL statements

Integrity Constraints are used to avoid redundancy in table.

To keep the integrity, accuracy and consistency of the data, Integrity constraints are used. Data integrity is take care of referential integrity. Referential integrity is a new concept used in all database systems. Integrity constraints will come in many ways like primary key , foreign key , unique constraints and other constraints.

The primary key field in a table particularly distinguishes every column or a record in the table of a database. The primary keys should have values that are unique and it can't have any values which are NULL.

Only one primary key is allowed in a table that might comprise of single or different fields. At the point when various fields are utilized as a primary key, they are known as a composite key.

If there is a primary key in any fields, One cannot have 2 records having the same values for that object.

The foreign key is a key utilized to connect two tables together and some of the time called as a referencing key.

Foreign Key is a segment or a blend of sections whose values coordinate a Primary Key in an alternate table.

Usually the primary key and foreign key establishes the relationship between two tables.

On the off chance that a table has a primary key characterized on any field(s), then you cannot have two records having the same estimation of that field(s).

In particular column, if two records are having the same value then lot of issues will come in DBMS. Unique constraint prevents this problem. In Customers table, two customers cannot have same customer id.

### Question 3

In vehicle table Registration Number is identified as primary Key which uniquely identifies a record in the table. In the same way In customer table Registration number is used as foreign key to enforce referential Integrity Which can be used to avoid duplication.

#### Description of Vehicle Table

S.No	Attribute	Datatype	Integrity Constraint
1	Registration_no	Number(6)	Primary Key
2	Make	Varchar2(20)	Not Null
3	Model	Varchar2(20)	
4	MOT	Date	Check Constraint

#### Description of Customer Table

S.No	Attribute	Datatype	Integrity Constraint
1	Cust_No	Number(6)	Primary Key
2	Cust_Name	Varchar2(20)	Not Null
3	Address	Varchar2(20)	
4	Phone	Number(10)	
5	Registration_No	Number(6)	Foreign Key

The customer table has serial number which is unique throughout the table, then the attributes contains the customer number, name, address and contact details such as address and phone number.

#### Question 4

(a) The creation of the Booking table

We can create booking table

Car\_no      varchar primary key

Start\_date    Date

End\_date      Date

Cus\_no      varchar foreign key

Create table booking(car\_no number(6), make varchar2(25),model varchar2(30),model number(6), start\_date date, end\_date date, cust\_no number(6) constraint custno\_fk foreign key reference customer(cust\_no));

We can create a table for booking with car\_no as primary key customer number who books the car as foreign key to enforce an integrity constraint to avoid redundancy.

(b) A new entry in the Vehicle table with:

RegNo: W 567 WDD

Make: Ford

Model: Mondeo

MOT: 23/3/2008

contractedServicer: 23 [1 mark]

insert into vehicle (RegNo., Make, Model, MOT, contractedServices) values('W567WDD', 'Ford', 'Mondeo', '23/3/2008', 23);



Here insert command is used to insert a booking detail in the table

It is conceivable to build relational algebra expressions utilizing various administrators like the utilization of arithmetic operators i.e., nesting of operators.

(c) An amendment to the above inserted record to correct the MOT date to 23/2/2008

update vehicle set MOT='23/2/2008' where MOT='23/3/2008'

Update command can be used to modify a record in the booking table (Eg. We can change the date).

(d) The names of all employees who had bookings on 23/12/2013

Select name from employee where booking\_date='23/12/2013'

Select command can be used to select the records from the table .Here where clause can be used check a condition.

(d) The number (just the number – not a list) of bookings authorised by team leader Bob Lockhart since 1st January 2013

Select \* from vehicle where start\_date='1/1/2013'

**Select command can be used to select the records from particular date.**

(e) Show in a single column the full names of all employees who have booked a vehicle for which the contracted servicer is the Garage called Fleet Services Ltd.

Select name from employee where contractedServicer ='Fleet Services Ltd';

Here select command with where condition is display records from booking table where  
contracedServicer ='Fleet Services Ltd';

(f) The following list (which shows the number of employees in each category)  
including the headings as shown

Category Number

T	1
R	3
O	3

**Select count(employee\_no) groupby category**

**The above command can be used to count no records Team leader, Regular and  
occasional users.**



**EssayCorp** 5 years ★★★★★

## Question 5

Database is nothing but collection of tables to store the data. Here four tables are designed like employee, vehicle, garage, customer details. In Employee table employee number is set as primary key. Category is set with check constraint to check whether the team leader 'T', or Regular user 'R', or Occasional User 'O'.

In Vehicle table Registration Number is set as Primary Key. MOT with date data type and set a check constraint to check the license details like expiry date.

Draw an Entity Relation Diagram to check functional dependencies in third normal form. Primary Key is nothing but which uniquely identifies a record in the table.

Foreign key is nothing referential integrity between a parent table and child table.

**DDL and DML both can be used effectively in database creation. Creation, Dropping and alteration are the major activities of the DDL language. Data selection, Data insertion, Data updating are the major activities in the DML programming language.**

A **data manipulation language (DML)** is a family of syntax elements similar to a computer programming **language** used for selecting, inserting, deleting and updating **data** in a database.

Ordinarily the clients of the database approach the centralized database. The distributed database has two applications they are local applications and global applications.

### **Local application**

In local applications the information is not recovered from different destinations.

### **Global application:**

In global application, the information is recovered from other social sites. Distributed databases are of two types they are Homogeneous and Heterogeneous database.

### **Homogeneous Distributed Database:**

Homogeneous distributed database is a solitary database in which all the hardware and software runs database things and happen at a single interface. It is exceptionally simple to design and to control the database as it has software that is indistinguishable and clients coordinate on preparing requests. All products convey its self-determination responsible for change construction and shows as single framework to the clients.

Some of the following conditions should be fulfilled for homogeneous distributed systems

1. The working framework should be comparative and perfect at every last area.
2. The utilization of information structure should be same and predictable among all areas.
3. The database applications utilized must be comparative and fitting at each area.

### **Heterogeneous Distributed Database**

A particular equipment, working framework, database administration frameworks and even information models for unmistakable databases are present in heterogeneous distributed database. The clients use particular construction that prompts issues in producing inquiry preparing and correspondence handling in this type of database system. While correspondence preparing the unmistakable locales gives restricted open doors for receiving correspondence process. Different software and hardware and information structures at different hubs are present for unmistakable hubs. The range is likewise improper, for example, one of the region will have current variant of relational database mechanics and another area will make traditional documents or might be old variants of database administration framework. Similarly, the utilized operating system by this database is distinctive as the areas have windows, whereas the other has UNIX and are free to one another. They even utilize their own particular hardware and software so that the customers of this database needs to login with database dialects. In the event that there is any change in the equipment of the framework, the PC might bring about adjusting the codes and word length and are financially powerless. The clients can read the information from other area yet they can't adjust or upgrade anything in different systems in this data base.

## References

- Brouard, F. and Soutou, C. (2005). *SQL*. Paris: Pearson Education.
- Celko, J. (2005). *Joe Celko's SQL for smarties*. Amsterdam: Morgan Kaufmann.
- Fehily, C. (2005). *SQL*. Berkeley, Calif.: Peachpit.
- Forta, B. (2000). *Sams teach yourself SQL in 10 minutes*. Indianapolis, Ind.: Sams.
- Harrington, J. (2002). *Relational database design clearly explained*. New York: Morgan Kaufmann Publishers.
- Harrington, J. (2003). *SQL*. Amsterdam: Morgan Kaufmann Publishers.
- McGrath, M. (2004). *SQL*. Southam: In Easy Steps.
- Rob, P. and Coronel, C. (2002). *Database systems*. Boston, MA: Course Technology.
- Stephens, R. (2009). *Beginning database design solutions*. Indianapolis, IN: Wiley Pub.
- Stephens, R. and Plew, R. (2001). *Database design*. Indianapolis, Ind.: Sams.
- Taylor, A. (2006). *SQL for dummies*. Hoboken, NJ: John Wiley.
- Teorey, T., Lightstone, S. and Nadeau, T. (2005). *Database modeling & design*. Amsterdam: Elsevier.
- Wilton, P. and Colby, J. (2005). *Beginning SQL*. Indianapolis, IN: Wiley Pub.