Chapter 3

Normalization

LEARNING OBJECTIVES

- *Normalization*
- *Anomalies*
- *First normal form*
- *Functional dependency*
- *Inference rules*
- *Second normal form*
- *Third normal form*
- *Higher normal forms (Boyce-Codd normal form)*
- *Fifth normal form*
- *Courses*

Normalization

Database design theory includes design standards called *normal forms*. The process of making data and tables match these standards is called *normalizing data* or *data normalization*. By normalizing data, we eliminate redundant information and organize table to make it easier to manage the data and make future changes to the table and database structure. This process removes the insertion, deletion, and modification anomalies. In normalizing your data, we usually divide large tables into smaller, easier to maintain tables. We can then use the technique of adding foreign keys to enable connections between the tables.

Data normalization is part of the database design process and is neither specific nor unique to any particular RDBMS. These are in order, such as first, second, third, Boyce-Codd, fourth, and fifth normal forms. Each normal form represents an increasingly stringent set of rules; that is, each normal form assumes that the requirements of the preceding forms have been met. Many relational database designers feel that, if their tables are in third normal form, most common design problems have been addressed. However, the higher-level normal forms can be of use and are included here.

Database normalization is the process of removing redundant data from tables to improve storage efficiency, data integrity and scalability.

1. In the relational model, methods exists for quantifying how efficient a database is, these classifications are called q'.
2. Normalization generally involves splitting existing tables into multiple ones, which must be rejoined (or) linked each time a query is issued.
3. Edgar F. Codd originally established three normal forms: 1NF, 2NF, 3NF. There are others also, but 3NF is widely considered to be sufficient for most applications, most tables when reaching 3NF are also in BCNF (Boyce–Codd normal form).

Table 1

Title	Author 1	Author 2	I SBN	Subject	Pages	Publisher
Database system concepts	Abraham Silber schatz	Henry F. Korth	0072958863	My SQL, computers	1160	McGraw-Hill
OS concepts	Abraham Silberschatz	Henry F. Korth	0471694665	Computers	990	McGraw-Hill

Problems:

1. This table is not very efficient with storage.
2. This design doesn't protect data integrity.
3. This table doesn't scale well.

Anomalies

An anomaly is a variation that differs in some way from what is said to be normal, with respect to maintaining a database.

1. The basic operations performed on Databases are Record insertion, Record updation, Record deletion.

2. It is desirable for these operations to be straight forward and efficient.
3. When relations are not fully normalized they exhibit anomalies.
4. The design goal of database is too easily to understand and to maintain.
5. Anomalies are problems that occur in un-normalized databases where all the data is stored in one table.

Types of anomalies

There are three types of anomalies that can arise in the database because of redundancy as follows:

1. Insertion anomaly
2. Deletion anomaly
3. Updation anomaly

Insertion anomaly An insertion anomaly occurs when particular attributes cannot be inserted into the database without the presence of other attributes.

Example: Consider the following table: Sales

Sales-Rep-Id	Name	Hire-Date	Client
1	Ana	1/1/2015	Madison
2	Sudha	2/4/2014	Peterson
3	Joey	3/2/2014	John
* New			

Insertion anomaly occurs in the above table which stores records for a company's sales representatives and the clients for whom they are responsible.

1. It is not possible to add records for newly hired Sales representatives until they have been assigned to one or more clients.
2. If we insert a record for newly hired, client column will be NULL, which is a required field for the table.
3. It is not possible to record newly hired in the table during training.

Deletion anomaly Deletion anomaly occurs when some particular attributes are lost because of the deletion of other attributes.

Example: Consider the following table 'course'.

S No	C No	S Name	Course
S41	C9201	John	Sales
S42	C9401	Brat	Finance
S40	C9201	Amit	Sales
S43	C9608	Arun	Accounts

Execute the following *SQL* query:

```
Delete    *
From      course
Where     S No = S43
```

If we delete a tuple where SNo = S43, he is the only (or) last student in the accounts department, we will lose data about student 'S43, Arun' as well as data about Accounts course that is 'C9608, Accounts'.

Updation anomaly An updation anomaly occurs when one or more instances of duplicated data are updated but not all.

Example: Consider the 'course' table given in the above example.

If we want to update course – No (Cno) of sales C9201 to C8686, in the course table.

1. It might happen that, the tuple with S No = S41 updated its CNo to C8686, but not the tuple with SNo = S43.
2. Inconsistency occurs in the table, because for the same course sales we have 2 different course Numbers.

Determining keys For a table 'R', its schema R consists of all attributes of R, we say X is a key to R if $X \rightarrow R$ means

X determines R
R is dependent upon X
If you know x then you know R

Example: Consider a relation schema R(ABCDE) and the functional dependencies:

$AC \rightarrow D$

$B \rightarrow E$

$DA \rightarrow B$

The closure of AC determines all the attributes present in Relation R, so the key for R is 'AC'.

$$
\begin{aligned}
AC^{+} = \quad & \{AC\} && \text{(self determination)} \\
& \{ACD\} && (AC \rightarrow D) \\
& \{ACDB\} && (DA \rightarrow B) \\
& \{ACDBE\} && (B \rightarrow E)
\end{aligned}
$$

$\therefore$ key = AC

Any attribute which does not appear on the right-hand-side of a given functional dependency appears in any one of the candidate keys.

1. From the above example, neither A (or) C appears in the right hand side of any functional dependency.

First Normal Form (1NF)

In Table 1, we have two violations of 1NF such as:

1. More than one author field and
2. Subject field contains more than one piece of information with more than one value in a single field; thus, it would be very difficult to search for all books on a given subject.

Table 2 *1NF table*

Title	Author	ISBN	Subject	Pages	Publisher
Database system concept	Abrahem Silbers Chatt	0072958863	My SQL	1160	McGraw-Hill
Database system concept	Henry K. Forth	0072958863	Computers	1160	McGraw-Hill
OS concepts	Henry K. Forth	0471694665	Computers	990	McGraw-Hill
OD concepts	Abraham Silber Schatz	0471694665	Computers	990	McGraw-Hill

In Table 2, we have two rows for a single book. Additionally, we would be violating the second NF. A better solution to the problem would be to separate the data into separate tables—an author table and a subject table to store our information, removing that information from the book table.

Table 3 *Subject table*

Subject–ID	Subject
1	My SQL
2	computers

Table 4 *Author table*

Author–ID	Last Name	First name
1	Silberschatz	Abraham
2	Korth	Henry

Table 5 *Book table*

ISBN	Title	Pages	Publisher
00729 58863	Database System Concepts	1160	McGraw-Hill
04716 94665	OS concepts	990	McGraw-Hill

Each table has a primary key, used for joining tables together when querying the data.

A table is in first normal form (1NF) if there are no repeating groups. A repeating group is a set of logically related fields or values that occur multiple times in one record. The sample tables below do not comply with first normal form. Look for fields that contain too much data and repeating group of fields.

EMPLOYEES_PROJECTS_TIME

A table with fields containing too much data.

Employee ID	Name	Project	Time
EN1-26	Sean O'Brien	30-452-T3, 30-457-T3, 32-244-T3	0.25, 0.40, 0.30
EN1-33	Amy Guya	30-452-T3, 30-382-TC, 32-244-T3	0.05, 0.35, 0.60
EN1-35	Steven Baranco	30-452-T3, 31-238-TC	0.15, 0.80
EN1-36	Elizabeth Roslyn	35-152-TC	0.90
EN1-38	Carol Schaaf	36-272-TC	0.75
EN1-40	Alexandra Wing	31-238-TC, 31-241-TC	0.20, 0.70

The example above is also related to another design issue, namely, that each field should hold the smallest meaningful value and that there should not be multiple values in a single field.

Why is this table design a problem?

There would be no way to sort by last names or to know which allocation of time belonged to which project.

EMPLOYEES_PROJECTS_TIME

Table 5 *A table with repeating groups of fields.*

Emp ID	Last Name	First Name	Project1	Time1	Project2	Time2	Project3	Time3
EN1-26	O'Brien	Sean	30-452-T3	0.25	30-457-T3	0.40	32-244-T3	0.30
EN1-33	Guya	Amy	30-452-T3	0.05	30-382-TC	0.35	32-244-T3	0.60
EN1-35	Baranco	Steven	30-452-T3	0.15	31-238-TC	0.80		
EN1-36	Roslyn	Elizabeth	35-152-TC	0.90				
EN1-38	Schaaf	Carol	36-272-TC	0.75				
EN1-40	Wing	Alexandra	31-238-TC	0.20	31-241-TC	0.70		

If an employee was assigned to a fourth project, you would have to add two new fields to the table. Also, it would be very difficult to total the amount of time devoted to a particular project.

The design problems addressed are very common, particularly among new designers who are accustomed to tracking data in a spreadsheet. Often, when building a spreadsheet, we arrange the data horizontally, laying it out across the spreadsheet. When designing tables, we have to think more vertically. Similar data belongs in the same column or field with a single value in each row.

Now we will take the table you saw above and redesign it so it will comply with first normal form.

Look at the repeating groups of data. Identify tables and fields that will hold this data without the repeating groups. Think vertically and remember that similar data belongs in the same field.

Enter the sample data from the table to make sure you don't have repeating groups. If necessary, include foreign key field(s) to connect the tables.

EMPLOYEES

EmployeeID	Last Name	First Name
EN1-26	O'Brien	Sean
EN1-33	Guya	Amy
EN1-35	Baranco	Steven
EN1-36	Roslyn	Elizabeth
EN1-38	Schaaf	Carol
EN1-40	Wing	Alexandra

PROJECTS_EMPLOYEES_TIME

Project Num	EmployeeID	Time
30-328-TC	EN1-33	0.35
30-452-T3	EN1-26	0.25
30-452-T3	EN1-33	0.05
30-452-T3	EN1-35	0.15
31-238-TC	EN1-35	0.80
30-457-T3	EN1-26	0.40
31-238-TC	EN1-40	0.20
31-241-TC	EN1-40	0.70
32-244-T3	EN1-33	0.60
35-152-TC	EN1-36	0.90
36-272-TC	EN1-38	0.75

Mark the primary key field(s) and foreign keys in each table. Shown below with * indicating the Primary key.

EMPLOYEES

EmployeeID	Last Name	First Name
EN1-26	O'Brien	Sean
EN1-33	Guya	Amy
EN1-35	Baranco	Steven
EN1-36	Roslyn	Elizabeth
EN1-38	Schaaf	Carol
EN1-40	Wing	Alexandra

PROJECTS_EMPLOYEES_TIME

Project Num	EmployeeID	Time
30-328-TC	EN1-33	0.35
30-452-T3	EN1-26	0.25
30-452-T3	EN1-33	0.05
30-452-T3	EN1-35	0.15
31-238-TC	EN1-35	0.80
30-457-T3	EN1-26	0.40
31-238-TC	EN1-40	0.20
31-241-TC	EN1-40	0.70
32-244-T3	EN1-33	0.60
35-152-TC	EN1-36	0.90
36-272-TC	EN1-38	0.75

If an employee was assigned to an additional project, it would involve merely adding a new record. Also, it would be much easier to search for a particular project number as they are all held in a single column.

Functional Dependency

A functional dependency is a relationship between fields so that the value in Field *A* determines the value in Field *B*, and there can be only one value in Field *B*. In that case, Field *B* is functionally dependent on Field *A*. Consider the following sample table:

Airport	City
National	Washington, DC
JFK	New York
LaGuardia	New York
Logan	Boston
Dulles	Washington, DC

Each airport name is unique and each airport can be in only one city. Therefore, City is functionally dependent on Airport. The value in the Airport field determines what the value will be in the City field (making Airport the determinant field) and there can be only one value in the City field. This does not need to work in the reverse. As shown in the table, a city can have more than one airport, so Airport is not functionally

dependent on City; the value in City does not necessarily determine what the value in Airport will be.

You will sometimes see a functional dependency written in this format:

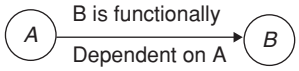
Determinant field(s) → Functionally dependent field

as in:

Airport → City

Functional dependency describes the relationship between attributes in a relation.

Example: If *A* and *B* are attributes of relation *R*, and *B* is functionally dependent on *A* (*A* → *B*) if each value of *A* is associated with one value of *B*.



Determinant refers to the attributed (or) group attributes on the left-hand side of the arrow of a functional dependency.

Inference Rules

The following inference rules IR 1 through IR 6 form a complete set for inferring functional and multi-valued dependencies from a given set of dependencies

Assume that all attributes are included in a 'universal' relation schema $R = \{A_1, A_2, \ldots A_N\}$ and that *X*, *Y*, *Z* and *W* are subsets of *R*.

IR 1 (reflexive rule): if $X \supseteq Y$, then $X \rightarrow Y$
IR 2 (Augmentation rule): $\{X \rightarrow Y\} = XZ \rightarrow YZ$
IR 3 (transitive rule): $\{X \rightarrow Y, Y \rightarrow Z\} = X \rightarrow Z$
IR 4 (complementation rule): $\{X \rightarrow \rightarrow Y\} = \{X \rightarrow \rightarrow (R - (X \cup Y))\}$
IR5 (augmentation rule for MVD's): if $X \rightarrow \rightarrow y$ and $W \rightarrow Z$ then $WX \supseteq YZ$
IR6 (transitive rule for MVD's):
$\{X \rightarrow \rightarrow Y, Y \rightarrow \rightarrow Z\} = X \rightarrow \rightarrow (Z - Y)$

Second Normal Form

A table is said to be in second normal form if it is in first normal form and each non-key field is functionally dependent on the entire primary key.

Look for values that occur multiple times in a non-key field. This tells us that we have too many fields in a single table.

Example: In the example below, see all the repeating values in the name and Project Title fields. This is an inefficient way to store and maintain data. In a well-designed database, the only data that is duplicated is in key fields used to connect tables. The presumption is that the data in key fields will rarely change, while the data in non-key fields may change frequently.

A table with a multifield primary key and repeating data in non-key fields

EmployeeID	Last Name	First Name	Project Number	Project Title
EN1-26	O'Brien	Sean	30-452-T3	STAR manual
EN1-26	O'Brien	Sean	30-457-T3	ISO procedures
EN1-26	O'Brien	Sean	31-124-T3	Employee handbook
EN1-33	Guya	Amy	30-452-T3	STAR manual
EN1-33	Guya	Amy	30-482-TC	Web Site
EN1-33	Guya	Amy	31-241-TC	New catalogue
EN1-35	Baranco	Steven	30-452-T3	STAR manual
EN1-35	Baranco	Steven	31-238-TC	STAR prototype
EN1-36	Roslyn	Elizabeth	35-152-TC	STAR pricing
EN1-38	Schaaf	Carol	36-272-TC	Order system
EN1-40	Wing	Alexandra	31-238-TC	STAR prototype
EN1-40	Wing	Alexandra	31-241-TC	New catalogue

If a ProjectTitle changed, we would have to edit it in several records. And what would happen in this table if the EmployeeID was part of the primary key and we wanted to add a new ProjectNum and ProjectTitle even though no employees had yet been assigned?

The primary key cannot contain a null value so you couldn't add the new project. Additionally, if a project ended and you wanted to delete it, you would have to delete the individual values because, if we deleted the records containing the titles and an employee was assigned to only that project, you would also delete that employee's record, something that we may not want to do.

In the above example, the asterisks indicate the fields that make up the primary key of this table as it now stands. A multifield primary key is necessary because neither the EmployeeID nor the ProjectNum fields contain unique values.

The reason there are repeated values in LastName, FirstName, and ProjectTitle is that these fields are dependent

on only part of the primary key. The value in EmployeeID determines what the value in LastName will be, but the value in ProjectNum has nothing to do with it. Similarly, the value in ProjectNum determines the value in ProjectTitle, but EmployeeID does not. These non-key fields relate to only part of the primary key. They are not functionally dependent on the entire primary key.

The solution to this lies in breaking the table into smaller tables that do meet second normal form. You will find that more tables are the solution to most problems encountered during data normalisation.

EMPLOYEES

EmployeeID	Last Name	First Name
EN1-26	O'Brien	Sean
EN1-33	Guya	Amy
EN1-35	Baranco	Steven
EN1-36	Roslyn	Elizabeth
EN1-38	Schaaf	Carol
EN1-40	Wing	Alexandra

EMPLOYEES_PROJECTS

EmployeeID	Project Num
EN1-26	30-452-T3
EN1-26	30-457-T3
EN1-26	31-124-T3
EN1-33	30-328-TC
EN1-33	30-452-T3
EN1-33	32-244-T3
EN1-35	30-452-T3
EN1-35	31-238-TC
EN1-36	35-152-TC
EN1-38	36-272-TC
EN1-40	31-238-TC
EN1-40	31-241-TC

Now we'll take the table above and design new tables that will eliminate the repeated data in the non-key fields.

1. To decide what fields belong together in a table, think about which field determines the values in other fields. Create a table for those fields and enter the sample data.
2. Think about what the primary key for each table would be and about the relationship between the tables. If necessary, add foreign keys or a junction table.
3. Mark the primary key for each table and make sure that you don't have repeating data in non-key fields.

PROJECTS

Projec tNum	Project Title
30-452-T3	STAR manual
30-457-T3	ISO procedures
30-482-TC	Web site
31-124-T3	Employee handbook
31-238-TC	STAR prototype
31-238-TC	New catalog
35-152-TC	STAR pricing
36-272-TC	Order system

Examine the tables to make sure there are no repeating values in non-key fields and that the value in each non-key field is determined by the value(s) in the key field(s). This removes the modification anomaly of having the repeated values.

Third Normal Form

A table is said to be in third normal form if it is in second normal form (2NF) and there are no transitive dependencies.

A transitive dependency is a type of functional dependency in which the value in a non-key field is determined by the value in another non-key field and that field is not a candidate key. Again, look for repeated values in a non-key field as in the following example.

A table with a single field primary key and repeating values in non-key fields.

Project Num	Project Title	Project Mgr	Phone
30-452-T3	STAR manual	Garrison	2756
30-457-T3	ISO procedures	Jacanda	2954
30-482-TC	Web site	Friedman	2846
31-124-T3	Employee handbook	Jones	3102
31-238-TC	STAR prototype	Garrison	2756
31-241-TC	New catalog	Jones	3102
35-152-TC	STAR pricing	Vance	3022
36-272-TC	Order system	Jacanda	2954

The phone number is repeated each time a manager's name is repeated. It is dependent on the manager, which is dependent on the project number (a transitive dependency).

The Project Manager field is not a candidate key, because the same person manages more than one project. Again, the solution is to remove the field with repeating data to a separate table.

Take the above table and create new tables to fix the problem.

1. Think about which fields belong together and create new tables to hold them.
2. Enter the sample data and check for unnecessarily (not part of primary key) repeated values.
3. Identify the primary key for each table and, if necessary, add foreign keys.

PROJECTS

Project Num	Project Title	Project Mgr
30-452-T3	STAR manual	Garrison
30-457-T3	ISO procedures	Jacanda
30-482-TC	Web site	Friedman
31-124-T3	Employee handbook	Jones
31-238-TC	STAR prototype	Garrison
31-241-TC	New catalog	Jones
35-152-TC	STAR pricing	Vance
36-272-TC	Order system	Jacanda

MANAGERS

Project Manager	Phone
Friedman	2846
Garrison	2756
Jacanda	2954
Jones	3102
Vance	3022

Reexamine your tables to make sure there are no unnecessarily repeating values in non-key fields and that the value in each non-key field is determined by the value(s) in the key field(s). In most cases, 3NF should be sufficient to ensure that your database is properly normalised.

Higher Normal Forms (Boyce–Codd Normal Form)

A table is in third normal form (3NF), and all determinants are candidate keys.

Boyce–Codd normal form (BCNF) can be thought of as a 'new' third normal form. It was introduced to cover situations that the 'old' third normal form did not address. The mean of a determinant (determines the value in another field) and candidate keys (qualify for designation as primary key). This normal form applies to situations where you have overlapping candidate keys.

If a table has no non-key fields, it is automatically in BCNF (Figure 1). Look for potential problems in updating existing data (modification anomaly) and in entering new data (insertion anomaly).

Imagine that we were designing a table for a college to hold information about courses, students, and teaching assistants. We have the following business rules:

1. Each course can have many students.
2. Each student can take many courses.
3. Each course can have multiple teaching assistants (TAs).
4. Each TA is associated with only one course.
5. For each course, each student has one TA.

Some sample data:

COURSES_STUDENTS_TA's

CourseNum	Student	TA
ENG101	Jones	Clark
ENG101	Grayson	Chen
ENG101	Samara	Chen
MAT350	Grayson	Powers
MAT350	Jones	O'Shea
MAT350	Berg	Powers

To uniquely identify each record, we could choose CourseNum + Student as a primary key. This would satisfy third normal form also because the combination of CourseNum and Student determines the value in TA. Another candidate key would be Student + TA. In this case, you have overlapping candidate keys (Student is in both). The second choice, however, would not comply with third normal form, because the CourseNum is not determined by the combination of Student and TA; it only depends on the value in TA. This is the situation that Boyce-Codd normal form addresses; the combination of Student + TA could not be considered to be a candidate key.

If we wanted to assign a TA to a course before any students enrolled, we couldn't because Student is part of the primary key. Also, if the name of a TA changed, would have to update it in multiple records. If assume have just these fields, this data would be better stored in three tables: one with CourseNum and Student, another with Student and TA, and third with CourseNum and TA.

COURSES

Course Num	Student
ENG101	Jones
ENG101	Grayson
ENG101	Samara
MAT350	Grayson
MAT350	Jones
MAT350	Berg

STUDENTS

Student	TA
Jones	Clark
Grayson	Chen
Samara	Chen
Grayson	Powers
Jones	O'Shea
Berg	Powers

TA's

*CourseNum	*TA
ENG101	Clark
ENG101	Chen
MAT350	O'Shea
MAT350	Powers

Figure 1 Tables that comply with BCNF.

Fourth Normal Form

A table is in Boyce-Codd normal form (BCNF) and there are no multi-valued dependencies.

A *multi-valued dependency* occurs when, for each value in field *A*, there is a set of values for field *B* and a set of values for field *C* but fields *B* and *C* are not related.

Look for repeated or null values in non-key fields. A multi-valued dependency occurs when the table contains fields that are not logically related. An often used example is the following table:

MOVIES

Movie	Star	Producer
Once Upon a Time	Julie Garland	Alfred Brown
Once Upon a Time	Mickey Rooney	Alfred Brown
Once Upon a Time	Julie Garland	Muriel Humphreys
Once Upon a Time	Mickey Rooney	Muriel Humphreys
Moonlight	Humphrey Bogart	Alfred Brown
Moonlight	Julie Garland	Alfred Brown

A movie can have more than one star and more than one producer. A star can be in more than one movie. A producer can produce more than one movie. The primary key would have to include all three fields, and so this table would be in BCNF. But you have unnecessarily repeated values, with the data maintenance problems that causes and you would have trouble with deletion anomalies.

The Star and the Producer really aren't logically related. The Movie determines the Star and the Movie determines the Producer. The answer is to have a separate table for each of those logical relationships: one holding Movie and Star and the other with Movie and Producer, as shown below:

STARS

*Movie	*Star
Once Upon a Time	Julie Garland
Once Upon a Time	Mickey Rooney
Moonlight	Humphrey Bogart
Moonlight	Julie Garland

PRODUCERS

*Movie	*Producer
Once Upon a Time	Alfred Brown
Once Upon a Time	Muriel Humphreys
Moonlight	Alfred Brown

Above, showing tables that comply with 4NF

Below is another example of a common design error, and it's easily spotted by all the missing or blank values.

PROJECTS_EQUIPMENT

Dept Code	Project Num	Project Mgr ID	Equipment	Property ID
IS	36-272-TC	EN1-15	CD-ROM	657
IS			VGA desktop monitor	305
AC	35-152-TC	EN1-15		
AC			Dot-matrix printer	358
AC			Calculator with tape	239
TW	30-452-T3	EN1-10	486 PC	275
TW	30-457-T3	EN1-15		
TW	31-124-T3	EN1-15	Laser printer	109
TW	31-238-TC	EN1-15	Handheld scanner	479
RI			Fax machine	775
MK			Laser printer	858
MK			Answering machine	187
TW	31-241-TC	EN1-15	Standard 19200 bps modem	386
SL			486 Laptop PC	772
SL			Electronic notebook	458

A table with many null values (Note: It also does not comply with 3NF and BCNF).

It is the same problem here because not all of the data is logically related. As usual, the answer is more tables: one to hold the information on the equipment assigned to departments (with PropertyID as the primary key) and another with projects and departments. We would now the business rules to know whether a project might involve more than one department or manager and be able to figure out the primary key. Assuming a project can have only one manager and be associated with only one department, the tables would be as follows:

EQUIPMENT

*Property ID	Equipment	DeptCode
657	CD-ROM	IS
305	VGA desktop monitor	IS
358	Dot-matrix printer	AC
239	Calculator with tape	AC
275	486 PC	TW
109	Laser printer	TW
479	Handheld scanner	TW
775	Fax machine	RI
858	Laser printer	MK
187	Answering machine	MK
386	Standard 19200 bps modem	TW
772	486 Laptop PC	SL
458	Electronic notebook	SL

PROJECTS_EQUIPMENT

Project Num	Project Mgr ID	Dept Code
36-272-TC	EN1-15	IS
35-152-TC	EN1-15	AC
30-452-T3	EN1-10	TW
30-457-T3	EN1-15	TW
31-124-T3	EN1-15	TW
31-238-TC	EN1-15	TW
31-241-TC	EN1-15	TW

Figure 2 Tables that eliminate the null values and comply with 4NF.

Fifth Normal Form

A table is in fourth normal form (4 NF) and there are no cyclic dependencies.

A *cyclic dependency* can occur only when you have a multifield primary key consisting of three or more fields. For example, let's say your primary key consists of fields *A*, *B*, and *C*. A cyclic dependency would arise if the values in those fields were related in pairs of *A* and *B*, *B* and *C*, and *A* and *C*.

Fifth normal form is also called *projection-join normal form*. A *projection* is a new table holding a subset of fields from an original table. When properly formed projections are joined, they must result in the same set of data that was contained in the original table.

Look for the number of records that will have to be added or maintained

Following is some sample data about buyers, the products they buy, and the companies they buy from.

BUYING

Buyer	Product	Company
Chris	Jeans	Levi
Chris	Jeans	Wrangler
Chris	Shirts	Levi
Lori	Jeans	Levi

Figure 3 A table with cyclic dependencies.

The primary key consists of all three fields. One data maintenance problem that occurs is that you need to add a record for every buyer who buys a product for every company that makes that product or they can't buy from them. That may not appear to be a big deal in this sample of two buyers, two products, and two companies ($2 \times 2 \times 2 = 8$ total records). But what if we went to 20 buyers, 50 products, and 100 companies ($20 \times 50 \times 100 = 100{,}000$ potential records)? It quickly gets out of hand and becomes impossible to maintain.

We might solve this by dividing this into the following two tables:

BUYERS

Buyer	Product
Chris	jeans
Chris	shirts
Lori	jeans

PRODUCTS

Product	Company
jeans	Wrangler
jeans	Levi
shirts	Levi

However, if you joined the two tables above on the Product field, it would produce a record not part of the original data set (it would say that Lori buys jeans from Wrangler). This is where the projection-join concept comes in.

The correct solution would be three tables:

BUYERS

*Buyer	*Product
Chris	jeans
Chris	shirts
Lori	jeans

PRODUCTS

*Product	*Company
jeans	Wrangler
jeans	Levi
shirts	Levi

COMPANIES

*Buyer	*Company
Chris	Levi
Chris	Wrangler
Lori	Levi

Figure 4 Tables that comply with 5NF.

When the first two tables are joined by Product and the result joined to the third table by Buyer and Company, the result is the original set of data.

EXERCISES

Practice Problems I

Directions for questions 1 to 20: Select the correct alternative from the given choices.

1. Consider the given functional dependencies

$A \rightarrow B$

$BC \rightarrow DE$

$AEF \rightarrow G$

Which of the following is true?

(A) Functional dependency $ACF \rightarrow DG$ implied by the set
(B) Functional dependency $ACF \rightarrow DG$ cannot be implied by the set
(C) Functional dependency $AB \rightarrow G$ implied by the set
(D) Both (*B*) and (*C*)

2. Consider the given relation

DNAME	DNO	MGRNO	LOCATION
RESEARCH	5	333	{BANGLORE,DELHI, HYDERABAD}
ADMINISTRATION	4	987	{CHENNAI}
EXECUTIVES	1	885	{HYDERABAD}

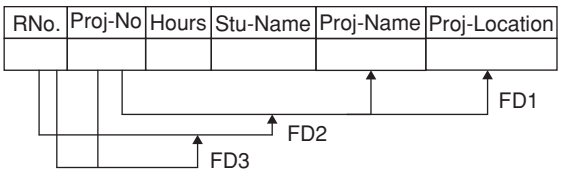
Department

The given relation is

(A) is not in 1NF (B) in 1NF
(C) in 2NF (D) in 3NF

3. Consider the given Relational scheme

Student-project

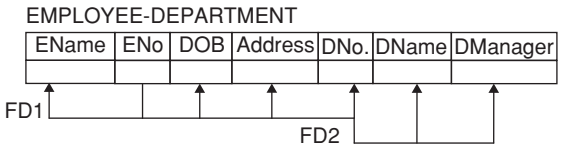


Which functional dependencies are violating 2NF property?

(A) FD1 (B) FD2
(C) FD3 (D) Both A and B

4. Consider the given relation

EMPLOYEE-DEPARTMENT



Which functional dependencies are violating 3NF ?

(A) FD1 (B) FD2
(C) Both (D) None of these

5. Consider the given relation $R(A, B, C, D)$and functional dependencies:

$FD = (AB \rightarrow C$

$C \rightarrow B$

$C \rightarrow D)$

Determine the key, prime attributes and non-prime attributes.

(A) $\{A\}, \{AB\}, \{CDE\}$
(B) $\{AB, AC\}, \{ABC\}, \{D\}$
(C) $\{AB, BC\}, \{ABC\}, \{D\}$
(D) $\{AB, AC\}, \{AB\}, \{D\}$

6. Consider the given relation and functional dependencies

$R(ABCDE)$

$FD = (ABD \rightarrow C$

$BC \rightarrow D$

$CD \rightarrow E)$

Determine the key, prime attributes, non-prime attributes and the normal form of the relation?

(A) $\{AB, AD\}, \{ABCD\}, \{E\}$
(B) $\{ABC, ABD\}, \{ABCD\}, \{E\}$
(C) $\{AB, AD\}, \{ABC\}, \{DE\}$
(D) $\{ABC, ABD\}, \{AB\}, \{CDE\}$

7. Consider the given relation and functional dependencies
$R(ABC)$
$FD = (AB \rightarrow C$
$C \rightarrow A)$
The relation is in which normal form?
(A) 1NF (B) 2NF
(C) 3NF (D) BCNF

8. Consider the given relation and its functional dependencies:
$R(ABCDE)$
$FD = (AB \rightarrow C$
$C \rightarrow E$
$B \rightarrow D$
$E \rightarrow A)$
The relation is further decomposed into two relations:
$R_1(BCD), R_2(ACE)$
(A) Decomposition is lossy and dependency preserving
(B) Decomposition is lossless and dependency preserving
(C) Decomposition is lossy and not dependency preserving
(D) Decomposition is lossless and not dependency preserving

9. Consider the following relational instance:

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

Which of the following functional dependencies are satisfied by the instance?
(A) $xy \rightarrow z$ and $z \rightarrow y$
(B) $yz \rightarrow x$ and $y \rightarrow z$
(C) $yz \rightarrow x$ and $x \rightarrow z$
(D) $xz \rightarrow y$ and $y \rightarrow x$

10. Consider the following functional dependencies:
DOB → Age
Age → Eligibility
Name → RNo
RNo → Name
CourseNo → CourseName
CourseNo → Instructor
(RNo, CNo) → Grade
The relation (RNo, Name, DOB, Age) is in which normal form?
(A) 1NF (B) 2NF
(C) 3NF (D) BCNF

11. Consider the given functional dependencies:
$AB \rightarrow CD$
$AF \rightarrow D$
$DE \rightarrow F$
$C \rightarrow G$
$F \rightarrow E$
$G \rightarrow A$
Which of the following is false?
(A) $\{CF\}^+ = \{ACDEFG\}$
(B) $\{BG\}^+ = \{ABCDG\}$
(C) $\{AF\}^+ = \{ACDEFG\}$
(D) $\{AB\}^+ = \{ABCDG\}$

12. What should be the key to make the given relation to be in BCNF? The dependencies for the following, 'Grades' relation are GRADES (student-Id, course#, semester#, Grade) student-Id, course#, semester# → Grade
(A) student-Id
(B) course#
(C) semester#
(D) student-Id, course#, semester #

13. What normal from is the following relation in?
STORE_ITEM (SKU, promotionID, vendor, style, price)
SKU, promotionID → vendor, style, price
SKU → vendor, style
(A) 1NF (B) 2NF
(C) 3NF (D) 4NF

14. What normal form is the following relation in?
Only H, I can act as the key
STUFF (H, I, J, K, L, M, N, O)
$H, I \rightarrow$ J,K,L
$J \rightarrow M$
$K \rightarrow N$
$L \rightarrow O$?
(A) 1NF (B) 2NF
(C) 3NF (D) BCNF

15. What normal form the following relation is in?
STUFF2(D, O, N, T, C, R, Y)
$D, O \rightarrow N, T, C, R, Y$
$C, R \rightarrow D$
$D \rightarrow N$?
(A) 1NF (B) 2NF
(C) 3NF (D) BCNF

16. The given table is in the BCNF form, convert it to the 4th normal form.

Employee	Skill	Language
Jones	Electrical	French
Jones	Electrical	German
Jones	Mechanical	French
Jones	Mechanical	German
Smith	Plumbing	Spanish

(A)

Employee	Skill

(B)

Employee	Language

(C)

Skill	Language

(D) Both A and B

17. For a database relation $x(a, b, c, d)$, where all the domains of a, b, c, d, include only atomic values, only the following FDs and those that can be inferred from them hold.

$a \rightarrow b, c \rightarrow d$

the relation is

(A) In 1st NF but not in 2nd NF
(B) In 2nd NF but not in 3rd NF
(C) In 2nd NF
(D) In 3rd NF

18. Which of the following FDs are satisfied by the instance from the below relation:

A	B	C
2	8	4
2	10	6
2	12	6
6	4	4

(A) AB → C and C → B
(B) BC → A and B → C
(C) BC → A and A → C
(D) AC → B and B → A

19. Consider the following database:

Course # → Title
Course # time → location
Emp – ID → T –Name salary
is in

(A) 3NF (B) 2NF
(C) 1NF (D) BCNF

20. Consider the following schema

$A = (w, x, y, z)$ and the dependencies are

$W \rightarrow X, X \rightarrow Y, Y \rightarrow Z$, and $Z \rightarrow W$

Let $A = (A_1$ and $A_2)$ be a decomposition such that $A_1 \cap A_2 = \phi$

The decomposition is

(A) In 1NF and in 2NF
(B) In 2NF and not in 3NF
(C) In 2NF and in 3NF
(D) Not in 2NF and in 3NF

Practice Problems 2

Directions for questions 1 to 20: Select the correct alternative from the given choices.

1. Integrity constraints ensures that changes made to the database by authorized users do not result in

(A) Loss of FDs
(B) Loss of keys
(C) Loss of tables
(D) Loss of data consistency

2. Relation $R = (\overline{A, B}, C, D)$ with AB as primary key. Choose one FD such that R should be in 1NF but not in 2NF

(A) $AB \rightarrow C$
(B) $AB \rightarrow D$
(C) $A \rightarrow D$
(D) $AB \rightarrow CD$

3. A normalized relation (1NF) can be retrieved from unnormalized relation by removing

(A) repeating groups (B) duplicate tuples
(C) transitive dependency (D) primary key

4. A relation will be in 2NF, if we

(A) remove repeating groups
(B) remove partial dependency
(C) remove transitive dependency
(D) have overlapping candidate key

5. Relation $R = (A, B, C, D)$ with AB as primary key, choose the FD so that R should be in 2NF but not in 3NF.

(A) $D \rightarrow C$ (B) $AB \rightarrow C$
(C) $AB \rightarrow D$ (D) $A \rightarrow B$

6. If a relation is in 2NF, then it can be in 3NF by removing

(A) repeating groups
(B) partial dependencies
(C) transitive dependencies
(D) overlapping dependencies

7. BCNF can be achieved from 3NF by removing

(A) repeating groups
(B) partial dependencies
(C) transitive dependencies
(D) overlapping dependencies

8. Which one of the following is not possible?
(A) Relation is in BCNF but not in 4NF
(B) Relation is in 3NF but not in BCNF
(C) Relation is in 2NF but not in 3NF
(D) Relation is in 3NF but not in 2NF

Common data for questions 9 and 10: Let R be a relation schema R (A, B, C, D);

$F = \{AB \rightarrow CD;\ C \rightarrow A\}$ F is the set of functional dependencies

9. How many prime attributes are there?
(A) 1 (B) 2
(C) 3 (D) 4

10. The highest normal form of the above relation is
(A) 1NF (B) 2NF
(C) 3NF (D) 4NF

Linked answer questions

11. For a given relation schema $R = \{A, B, C, D, E\}$
$A \rightarrow BC$
$CD \rightarrow E$
$B \rightarrow D$
$E \rightarrow A$
Which of the following is not a candidate key?
(A) A (B) B
(C) E (D) BC

12. For the above answer, what is the closure?
(A) BD (B) ABC
(C) $ABCDE$ (D) BC

13. Consider the following functional dependencies:
$A \rightarrow B$
$C \rightarrow D$
$B \rightarrow E$
$F \rightarrow A$

The relation (A, B, C, D) is
(A) in second normal form, but not in third normal form
(B) in third normal form, but not in $B\ C\ N\ F$
(C) in $B\ C\ N\ F$
(D) None of the above

Common data for questions 14 and 15:

$R = (A, B, C, D, E, F)$
FDs = $A \rightarrow B$
$C \rightarrow$ DF
$AC \rightarrow E$
$D \rightarrow F$

14. Determine the key from the given FDs:
(A) AB (B) AC
(C) ACB (D) ACD

15. Decompose the FDs into 2NF
(A) $R_1(AB)\ R_2(CDF)\ R_3(ACE)$
(B) $R_1(AB)\ R_2(CDEF)$
(C) $R_1(ABC)\ R_2(CDF)$
(D) $R_1(AB)\ R_2(CD)\ R_3(EF)$

16. For a database relation $x(a, b, c, d)$, where all the domains of a, b, c, d, include only atomic values, only the following FDs and those that can be inferred from them hold.

$a \rightarrow b$, $c \rightarrow d$

The relation is decomposed into $R_1(ab)$, $R_2(cd)$. Which of the following is true, The decomposition
(A) is dependency preserving
(B) is not dependency preserving
(C) is loss less
(D) Both A and C

17. Which of the following FDs are satisfied by the instance from the below relation?

A	B	C
4	12	8
4	14	10
4	16	10
10	8	8

(A) $AB \rightarrow C$ and $C \rightarrow B$
(B) $BC \rightarrow A$ and $B \rightarrow C$
(C) $BC \rightarrow A$ and $A \rightarrow C$
(D) $AC \rightarrow B$ and $B \rightarrow A$

18. Indicate which of the following statements are false: 'A relational database, which is in 3NF still have undesirable data redundancy because there may exist.
(A) Below all
(B) Non trivial FDs involving prime attributes on the right side.
(C) Non-trivial FDs involving prime attributes on the left side
(D) Non-trivial FDs involving only prime attributes

19. Consider the following database:
SOFTWARE (software-vendor, product, Release-date, systemReq, warranty)
FD: (software-vendor, product, Releasedate) → system Req, price, Warranty.
Which of the following are non–prime attributes?
(A) SystemReq
(B) Price
(C) Warranty
(D) All the above

20. Consider a relation schema R (A, B, C, D, E, X, Y) with the following FDs
$F = \{0 \rightarrow A, XD \rightarrow C, DA \rightarrow B, A \rightarrow X, XE \rightarrow B, E \rightarrow A, B \rightarrow D, DA \rightarrow B, EB \rightarrow C, AB \rightarrow C, Y \rightarrow B, C \rightarrow B\}$ is in
(A) 2NF
(B) 3NF
(C) 4NF
(D) BCNF

Previous Years' Questions

1. Which one of the following statements is false? **[2007]**
(A) Any relation with two attributes is in BCNF
(B) A relation in which every key has only one attribute is in 2NF
(C) A prime attribute can be transitively dependent on a key in a 3NF relation.
(D) A prime attribute can be transitively dependent on a key in a BCNF relation.

2. Consider the following relational schemas for a library database:

Book (Title, Author, Catalog_ no, Publisher, Year, Price)

Collection (Title, Author, Catalog_ no)

with the following functional dependencies:

I. Title Author → Catalog_no
II. Catalog_no → Title Author Publisher Year
III. Publisher Title Year → Price

Assume {Author, Title} is the key for both schemas. Which of the following statements is true? **[2008]**
(A) Both Book and Collection are in BCNF
(B) Both Book and Collection are in 3NF only
(C) Book is in 2NF and Collection is in 3NF
(D) Both Book and Collection are in 2NF only

3. The following functional dependencies hold for relations $R(A, B, C)$ and $S(B, D, E)$
$B \rightarrow A$,
$A \rightarrow C$

The relation R contains 200 tuples and the relation S contains 100 tuples. What is the maximum number of tuples possible in the natural join $R \bowtie S$? **[2010]**
(A) 100 (B) 200
(C) 300 (D) 2000

4. Which of the following is true? **[2012]**
(A) Every relation in 3NF is also in BCNF
(B) A relation R is in 3NF if every non-prime attribute of R is fully functionally dependent on every key of R
(C) Every relation in BCNF is also in 3NF
(D) No relation can be in both BCNF and 3NF

Common data questions 5 and 6: Relation R has eight attributes ABCDEFGH, Fields of R contain only atomic values.

$F = \{CH \rightarrow G, A \rightarrow BC, B \rightarrow CFH, E \rightarrow A, F \rightarrow EG\}$ is a set of functional dependencies (FDs) so that F^+ is exactly the set of FDs that hold for R.

5. How many candidate keys does the relation R have? **[2013]**
(A) 3 (B) 4
(C) 5 (D) 6

6. The relation R is **[2013]**
(A) in 1NF, but not in 2NF
(B) in 2NF, but not in 3NF
(C) in 3NF, but not in BCNF
(D) in BCNF

7. Assume that in the suppliers relation above, each supplier and each street within a city has a unique name, and (sname, city) forms a candidate key. No other functional dependencies are implied other than those implied by primary and candidate keys. Which one of the following is true about the above schema? **[2009]**
(A) The schema is in BCNF
(B) The schema is in 3NF but not in BCNF
(C) The schema is in 2NF but not in 3NF
(D) The schema is not in 2NF

8. Consider the relation scheme $R = (E, F, G, H, I, J, K, L, M, N)$ and the set of functional dependencies $\{\{E, F\} \rightarrow \{G\}, \{F\} \rightarrow \{I, J\}, \{E, H\} \rightarrow \{K, L\}, \{K\} \rightarrow \{M\}, \{L\} \rightarrow \{N\}\}$ on R. What is the key for R? **[2014]**
(A) $\{E, F\}$ (B) $\{E, F, H\}$
(C) $\{E, F, H, K, L\}$ (D) $\{E\}$

9. Given the following two statements:

S_1: Every table with two single-valued attributes is in 1NF, 2NF, 3NF and BCNF

S_2: $AB \rightarrow C, D \rightarrow E, E \rightarrow C$ is a minimal cover for the set of functional dependencies $AB \rightarrow C, D \rightarrow E, AB \rightarrow E, E \rightarrow C$

Which one of the following is correct? **[2014]**
(A) S_1 is true and S_2 is false
(B) Both S_1 and S_2 are true
(C) S_1 is false and S_2 is true
(D) Both S_1 and S_2 are false

10. The maximum number of super-keys for the relation schema R (E, F, G, H) with E as the key is _____. **[2014]**

11. A *prime attribute* of a relation scheme R is an attribute that appears **[2014]**
(A) in all candidate keys of R
(B) in some candidate key of R
(C) in a foreign key of R
(D) only in the primary key of R

12. Consider an entity-Relationship (ER) model in which entity sets E_1 and E_2 are connected by an m:n relationship R_{12}. E_1 and E_3 are connected by a 1:n (1 on the side of E_1 and n on the side of E_3) relationship R_{13}.

E_1 has two single-valued attributes a_{11} and a_{12} of which a_{11} is the key attribute. E_2 has two single-valued

attributes a_{21} and a_{22} of which a_{21} is the key attribute. E_3 has two single-valued attributes a_{31} and a_{32} of which a_{31} is the key attribute. The relationships do not have any attributes.

If a relational model is derived from the above ER model, then the minimum number of relations that would be generated if all the relations are in 3 NF is _____. **[2015]**

13. Consider the relation $X(P, Q, R, S, T, U)$ with the following set of functional dependencies

```
F = {
    {P, R} → {S, T}
    {P, S, U} → {Q, R}
    }
```

Which of the following is the trivial functional dependency in $F+$, where $F+$ is closure of F? **[2015]**

(A) $\{P, R\} \rightarrow \{S, T\}$
(B) $\{P, R\} \rightarrow \{R, T\}$
(C) $\{P, S\} \rightarrow \{S\}$
(D) $\{P, S, U\} \rightarrow \{Q\}$

14. A database of research articles in a journal uses the following schema. **[2016]**

(VOLUME, NUMBER, STARTPAGE, ENDPAGE, TITLE, YEAR, PRICE)

The primary key is (VOLUME, NUMBER, STARTPAGE,ENDPAGE) and the following functional dependencies exist in the schema.

(VOLUME, NUMBER, STARTPAGE, ENDPAGE) → TITLE

(VOLUME, NUMBER) → YEAR

(VOLUME, NUMBER,

STARTPAGE,ENDPAGE) → PRICE

The database is redesigned to use the following schemas.

(VOLUME, NUMBER, STARTPAGE, ENDPAGE, TITLE, PRICE)

(VOLUME, NUMBER, YEAR)

Which is the weakest normal form that the new database satisfies, but the old one does not?

(A) 1NF (B) 2NF
(C) 3NF (D) BCNF

15. Consider the following database table water_ schemes: **[2016]**

water_schemes

scheme_no	District name	Capacity
1	Ajmer	20
1	Bikaner	10
2	Bikaner	10
3	Bikaner	20
1	Churu	20
2	Churu	20
1	Dungargarh	10

The number of tuples returned by the following SQL query is _____ .

with total (name, capacity) **as**
select district _ name, **sum** (capacity)
from water _ schemes
group by district _ name
with total _avg (capacity) **as**
select avg (capacity)
from total
select name
from total, total _ avg
where total . capacity ≥ total_avg. capacity

16. The following functional dependencies hold true for the relational schema $R\{V, W, X, Y, Z\}$:

$$V \rightarrow W$$
$$VW \rightarrow X$$
$$Y \rightarrow VX$$
$$Y \rightarrow Z$$

Which of the following is irreducible equivalent for this set of set of functional dependencies? **[2017]**

(A) $V \rightarrow W$, $V \rightarrow X$, $Y \rightarrow V$, $Y \rightarrow Z$
(B) $V \rightarrow W$, $W \rightarrow X$, $Y \rightarrow V$, $Y \rightarrow Z$
(C) $V \rightarrow W$, $V \rightarrow X$, $Y \rightarrow V$, $Y \rightarrow X$, $Y \rightarrow Z$
(D) $V \rightarrow W$, $W \rightarrow X$, $Y \rightarrow V$, $Y \rightarrow X$, $Y \rightarrow Z$

17. Consider the following tables T1 and T2.

T1

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

T2

R	S
2	2
8	3
3	2
9	7
5	7
7	2

In table T1, **P** is the primary key and **Q** is the foreign key referencing **R** in table T2 with on-delete cascade and on-update cascade. In table T2, **R** is the primary key and **S** is the foreign key referencing **P** in table Tl

with on-delete set NULL and on-update cascade. In order to delete record ⟨3, 8⟩ from table T1, the number of additional records that need to be deleted from table T1 is _____. **[2017]**

18. Consider the following four relational schemas. For each schema, all non-trivial functional dependencies are listed. The underlined attributes are the respective primary keys.

Schema I:

Registration (rollno, courses)

Field 'courses' is a set-valued attribute containing the set of courses a student has registered for.

Non-trivial functional dependency:

Rollno → courses

Schema II:

Registration (rollno, courseid, email)

Non-trivial functional dependencies:

Rollno, courseid → email

email → rollno

Schema III:

Registration (rollno, courseid, marks, grade)

Non-trivial functional dependencies:

Rollno, courseid → marks, grade

Marks → grade

Schema IV:

Registration (rollno, courseid, credit)

Non-trivial functional dependencies:

Rollno, courseid → credit

Courseid → credit

Which one of the relational schemas above is in 3NF but not in BCNF? **[2018]**

(A) Schema I
(B) Schema II
(C) Schema III
(D) Schema IV

Answer Keys

Exercises

Practice Problems 1

1. A	**2.** A	**3.** D	**4.** B	**5.** B	**6.** B	**7.** C	**8.** D	**9.** B	**10.** A
11. C	**12.** D	**13.** A	**14.** B	**15.** A	**16.** D	**17.** A	**18.** B	**19.** C	**20.** C

Practice Problems 2

1. D	**2.** C	**3.** A	**4.** B	**5.** A	**6.** C	**7.** D	**8.** D	**9.** C	**10.** C
11. B	**12.** A	**13.** D	**14.** B	**15.** A	**16.** A	**17.** C	**18.** C	**19.** D	**20.** B

Previous Years' Questions

1. D	**2.** C	**3.** A	**4.** C	**5.** B	**6.** A	**7.** B	**8.** B	**9.** A	**10.** 8
11. B	**12.** 4	**13.** C	**14.** B	**15.** 2	**16.** A	**17.** 0	**18.** B		