PG & REASEARCH DEPARTMENT OF COMPUTER SCIENCE

GOVERNMENT ARTS COLLEGE (GRADE-I) – ARIYALUR

DATABASE SYSTEMS

COURSE CODE: 16SCCCS4

Normalization of Database Tables

Database Tables & Normalization:

1. In Database designed process, the table is the basic building block.
2. The ER model gives good table structure. But it is possible to create poor table structure. Even in a good database structure design.

Def:

Normalization is an Analysis of functional dependency between the attributes of a relation. It reduces the complex user view into set of stable sub groups or fields.

The normalization process is used to create a good table structure to minimize data redundancy.

Normalization works through a series of stages called normal form.

The first three stages are

- First Normal Form(1NF)
- Second Normal Form(2NF)
- Third Normal Form(3NF)

Business Databases: Business databases are sufficient to normalize to 2NF or 3NF. The other stages are

- Boyce Code Normal Form (BCNF)
- Fourth Normal Form(4NF)
- Fifth Normal Form (5NF)

Normalization is a very important in database design .Generally the higher normal forms, the more relational join operations required to produce a specific output.

Therefore occasionally we accepted to denormalize some positions of the database to increase the efficiency.

DeNormalization produces a lower normal form i.e., 3NF will be connected into 2NF will be converted into 1NF.

The need for Normalization:

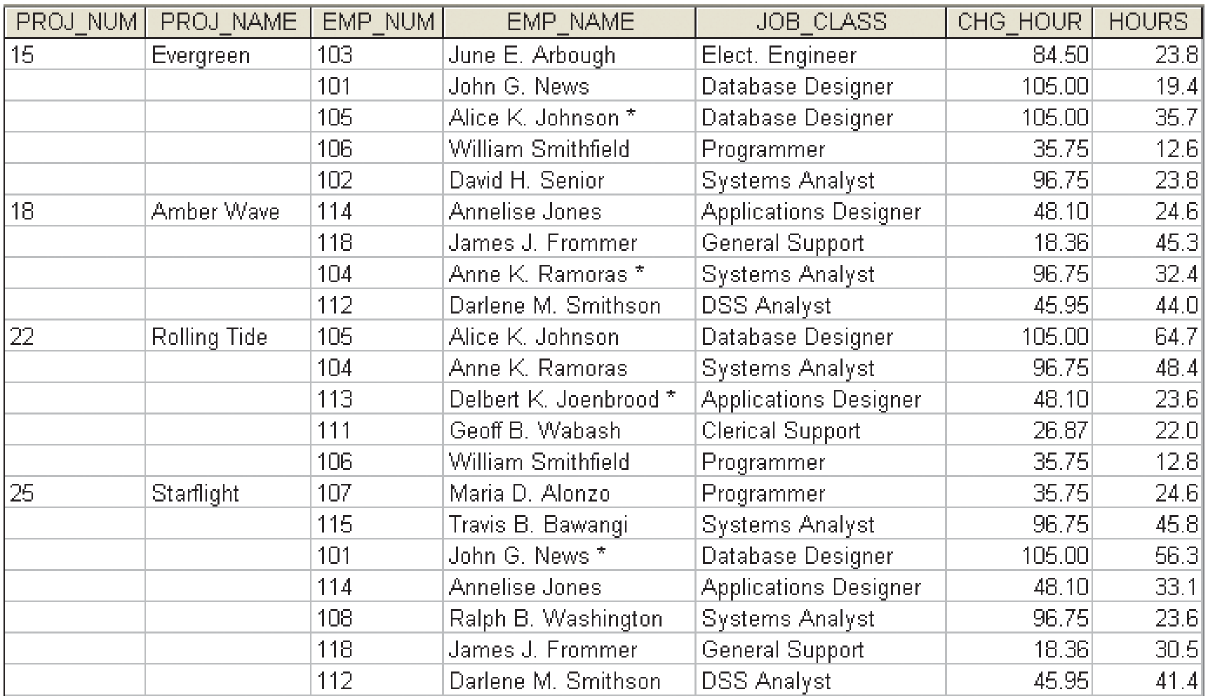
Consider the Database activities of a construction company that manages several building projects. Each Project has its own project number and project name, employee assigned to it and soon. Each employee has employee number, employee name, classification etc.

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS	Total_Charge
15	Evergreen	103	June E .Arbough	Elect.Engineer	84.50	23.8	2011.1
		101	Swetha	Database designer	105.00	19.4	2037
		105	Lakshmi	Database designer	105.00	35.7	3748.5
		106	Durga	Programmer	35.75	12.6	450.45
		102	Ram	Systems Analyst	96.75	20.8	2012.4
SubTotal 10259.45							
18	Amber wave	114	Harika	Applications designer	48.10	25.6	1231.36
		118	Ganesh	General support	18.36	45.3	831.708
		104	Sri	Systems analyst	96.75	32.4	3134.7
		112	Hari	DSS Analyst	45.95	44.0	2021.8
						Sub Total	7219.568
22	Rolling Tide	105	Sruthi	Database designer	105.00	64.7	6793.5
		104	Raju	Systems analyst	26.75	48.4	1294.7
		113	Ravi	Application designer	48.10	23.6	1135.16
		111	Ramesh	Clerical support	26.87	22.0	591.14
		106	Rao	Programmer	35.75	12.8	457.6
						Sub Total	10272.1
25	Starflight	107	Rekha	Programmer	35.75	24.6	879.45
		115	Rani	System analyst	96.75	45.8	4431.15
		101	John	Database designer	105.00	56.3`	5911.5

		114	Manikanta	Applications designer	48.10	33.1	1592.11
		108	Nalini	System analyst	96.75	23.6	2283.3
		118	James	General secretary	18.36	30.5	559.98
		112	P J	DSS Analyst	45.95	41.4	1902.33
						Sub Total	17559.82
						Total Amount	45310.94

The Easiest way to generate the required report to create a table that table has some fields of the Report.

Table_Name : Construction_Company



PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
		101	John G. News	Database Designer	105.00	19.4
		105	Alice K. Johnson *	Database Designer	105.00	35.7
		106	William Smithfield	Programmer	35.75	12.6
		102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
		118	James J. Frommer	General Support	18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
		111	Geoff B. Wabash	Clerical Support	26.87	22.0
		106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	96.75	45.8
		101	John G. News *	Database Designer	105.00	56.3
		114	Annelise Jones	Applications Designer	48.10	33.1
		108	Ralph B. Washington	Systems Analyst	96.75	23.6
		118	James J. Frommer	General Support	18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	45.95	41.4

An Employee can be assigned more than one project.

For example: Employee number 104 has been assigned to two project .Therefore knowing the project _no and employee no will find the job classification and hours worked. Therefore project_No and emp_no will be taken as primary key.

The above structure of the table has the following deficiency

1. The project _no is a part of primary key. But it contains null values.
2. The table entries invites data inconsistency for example job classification value Electrical_Engineer might be entered.Elec_engi ,EE
3. The table displays data redundancy.

Update Anomalies: Modify the job class for Employee_No 105 requires many alternatives.

Insertion Anomalies: To complete a row definition of a new employee must be assigned to a project. If the employee is not assigned, a dummy project must be created to complete the row.

Deletion Anomalies: Suppose only one employee is associated with a project, if that employee leaves the company and the employee data are deleted, the project information will also be deleted.

The above deficiency of table structure appears to work, the report gives different results depending on data.

Normalization Process: The most common Normal forms and their characteristics are

1. **First Normal Form (1NF):** A Relation is said to be in first normal form if it is already in un normalized form and it has no repeating group.
2. **Second Normal Form (2NF):** A Relation is said to be in second normal form if it is already in first normal form and it has no partial dependency.
3. **Third Normal Form (3NF):** A Relation is said to be in third Normal form if it is already in second normal form and it has no transitive dependency.
4. **Boyce code Normal Form(BCNF):** A Relation is said to be in Boyce code Normal form if it is already in third normal form and every determinant is a candidate key.
5. **Fourth Normal Form (4NF):** A Relation is said to be in fourth normal form if it is already in Boyce code normal form and it has no multi valued dependency.
6. **Fifth Normal Form(5NF):** A Relation is said to be fifth normal form if it is already in fourth normal form and it has no loss less decompose.

Eg: Normalization of construction company Report

PROJ_NUM	PROJ_NAME	EMP_ NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS	Total_Charge
15	Evergreen	103	June E .Arbough	Elect.Engineer	84.50	23.8	2011.1
		101	Swetha	Database designer	105.00	19.4	2037
		105	Lakshmi	Database designer	105.00	35.7	3748.5
		106	Durga	Programmer	35.75	12.6	450.45
		102	Ram	Systems Analyst	96.75	20.8	2012.4
SubTotal 10259.45							
18	Amber wave	114	Harika	Applications designer	48.10	25.6	1231.36

		118	Ganesh	General support	18.36	45.3	831.708
		104	Sri	Systems analyst	96.75	32.4	3134.7
		112	Hari	DSS Analyst	45.95	44.0	2021.8
						Sub Total	7219.568
22	Rolling Tide	105	Sruthi	Database designer	105.00	64.7	6793.5
		104	Raju	Systems analyst	26.75	48.4	1294.7
		113	Ravi	Application designer	48.10	23.6	1135.16
		111	Ramesh	Clerical support	26.87	22.0	591.14
		106	Rao	Programmer	35.75	12.8	457.6
						Sub Total	10272.1
25	Starflight	107	Rekha	Programmer	35.75	24.6	879.45
		115	Rani	System analyst	96.75	45.8	4431.15
		101	John	Database designer	105.00	56.3`	5911.5
		114	Manikanta	Applications designer	48.10	33.1	1592.11
		108	Nalini	System analyst	96.75	23.6	2283.3
		118	James	General secretary	18.36	30.5	559.98
		112	P J	DSS Analyst	45.95	41.4	1902.33
						Sub Total	17559.82
						Total Amount	45310.94

The construction company report is represented in the form of relation. The relation named as CONSTRUCTION_COMPANY this is in un normalized form as shown below

CONSTRUCTION_COMPANY(Proj_No,Proj_Name,(Emp_No,Emp_Name, Job_Classification, Charge_Per_Hour, Hours_ Billed)) -----→ (1)

The field Total charge, SUB TOTAL,GRAND TOTAL are not included in the relation because they are derived Attribute.

First Normal Form:

In Relation (1), the fields in the inner most set of parenthesis put together is known as repetating group. This will result in redundancy of data for the first two relations remove the repetating group. Hence the relation 1 is subdivided into two relations to remove repeating group

PROJECT(proj_No,proj_Name) ----------> (2)

PROJECT_EMP(Proj_No, Emp_No,Emp_Name,Job_Class,Charge_Per_Hour,Hours_Billed) ------ →(3)

Now above relation (2) & (3) are in 1NF. In relation (3) Proj_No , Emp_No jointly serve as key field.

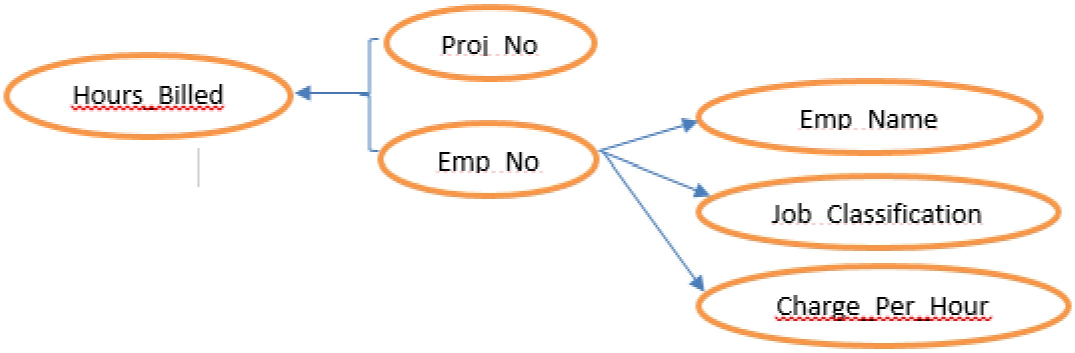
Second Normal Form:

Definition of Partial Dependency:

Non key attribute are depending on the part of the composite primary key then it is said to be Partial Dependency.

In Relation 2 the number key fields is only one and hence there is no scope for partial dependency the absence of partial dependency in relation 2 takes it 2NF without any modification.

The dependency diagram of relation 3 is shown below



In the above diagram **Hours_Builled** depends on Project_No and Emp_No but the remaining non key fields **(**Emp_Name,Job_Class,Charge_per_Hour)depends on Emp_No this situation is an example of 2nd normal form .Hence the relation 3 is divided into 2 relations.

Assignment(Proj_No, Emp_No, Hours_Billed)------------→(4)

Emp_Job(Emp_No,Emp_Name,Job_Class,Charge-Per_Hour)--------------→(5)

The Relations (4) & (5) are in 2NF

Third Normal Form:

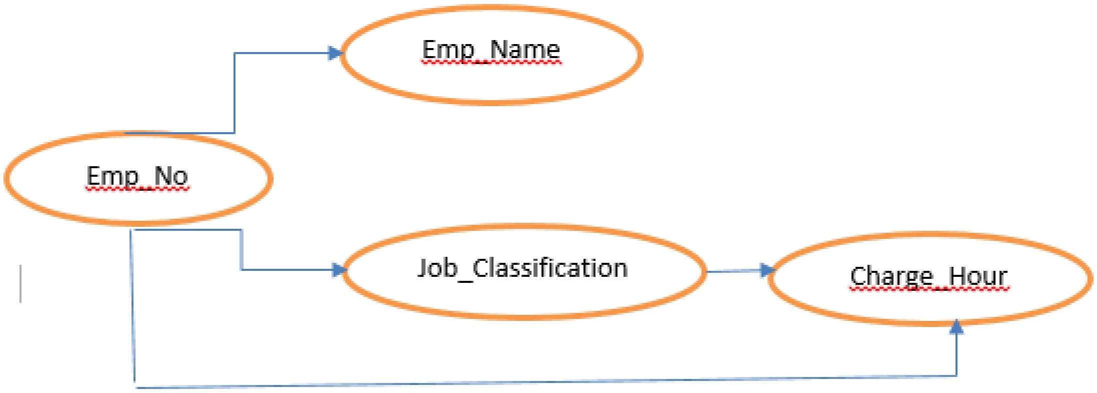
Transitive Dependency:

If one non prime attribute is determines the other non prime attribute then it is called as transitive dependency..

In Relation (2) there is only one non key field. This means that it has no transitive dependency. Hence Relation (2) can be treated as 3NF without any modification similarly in relation (4) there is only one non key field. This means that it has no transitive dependency. Hence relation (4) can be treated as 3Nf without any modification.

In Relation(5) Charge_Per_Hour depends on Job_Classification this means that relation (5) has transitive dependency. The dependency diagram for the relation (5) is shown below.

Diagram



Hence relation (5) is sub divided into two relations. Relation (6) and relation (7) as shown below.

Job(Job_Class, Charge_Per_Hour)----------------→(6)

Emp(Emp_No,Emp_Name,Job_Class)-----------→(7)

For a practical application it is sufficient to normalized up to either 2NF or 3NF

Hence, the process of normalization is stopped and the final 3NF relations of construction company as shown below.

Project(Proj_No,Proj_Name)-------------------------------→(1)

Assignment(Proj_No, Emp_No, Hours_Billed)-----------→(2)

Emp(Emp_No,Emp_Name,Job_Classification)-----------→(3)

Job(Job_Classification, Charge_Per_Hour)---------------→(4)

Improving Design:

How to improve the design of the database?

1. **Evaluate primary key Assignment:**

   Each time a new employee is entered into the employee table, A job class value must be entered. Unfortunately the data entry in the job class contains an error, that lead to referential integrity violation. For example entering "DB Designer"instead of Database Designer for the Job_class attribute in the Employee table will trigger such a violation . Therefore it is better to add job code attribute in job relation and employee relation.

   Emp(Emp_No,Emp_Name,Job_Code)
   Job(Job_Code,Job_Classification,Charge_Per_Hour)

2. **Evaluate Naming conversions:**

   In the job relation job classification will be changed to Job_Description and Charge_Per_Hour will be charged to Job_Charg_Hour. In the assignment relation Hours_Billed will be changed to Assign_Hours_Billed.

   Job(Job_Code,Job_Description,Job_Chg_Hour)
   Assignment(Proj_No,Emp_No,Assign_Hours_Billed)

3. **Refine Attribute Atomicity:**

   An Automatic Attribute is one that cannot be further sub divided such an attribute is said to be automaticity. In employee table the attribute Emp_Name is not a automaticity because it is further sub divided as Emp_LName,Emp_FName,Emp_Init. These attributes are added to the employee table.

   Emp(Emp_No,Emp_LName,Emp_FName,Emp_Init,Job_Code)

4. **Identify New Attribute:**

   If the employee table used in real world environment several other attributes will be added. For example Social_Security_Number, Date_of_Joining, Date_of_Birth,Hire_Date,Gross_Salary,Net_salary etc will be added to improve relation.

Emp(Emp_No,Emp_LName,Emp_FName,Emp_Init,Hire_Data,Gross_Salary,Job_Code)

5. **Identify New Relationship:**

When we create a new relationship between the table it will not produce unnecessary duplication. Then we create a new relationship.

6. **Refine primary Keys:**

The combination of Emp_No and Proj_No is a primary key in the Assignment table. For example if we add a assigned hours more than one time for a particular project then it violates the primary key constraints.

To avoid the violation to add additional attribute Assign_Date to the assignment table. If we want to add assigned hours for a particular project more than one time in the same day then it will violates the primary key constraints. The same data entry gives no problem when Assign_No is used as a primary key in the Assignment relation.

Assignment(Assign_No,Assign_Date,Proj_No,Emp_No,Assig_Hour_Billed)

7. **To Maintain Historical Accuracy:**

It is assumed that the Job_Chg_Hour will change over time. The changes to each project were billed by multiplying the hours worked on the project in the assignment table by the Job_Chg_Hour in the job table. Those changes would always show the current **change_ per_hour** stored in the job table rather than the job charge hour that was in effect at the time of assignment. Because of that we are adding an attribute **Assign_Chg_Hour** to the Assignment table.

Assignment(Assign_No,Assign_Date,Proj_No,Emp_No,Assig_Hour_Billed,Assign_Chg_Hour)

8. **Evaluate using Derived Attribute:**

The derived attribute Assign_Charge is added to the Assignment relation and Assign_charge is updated by multiplying with Assign_Chg_Hour with the

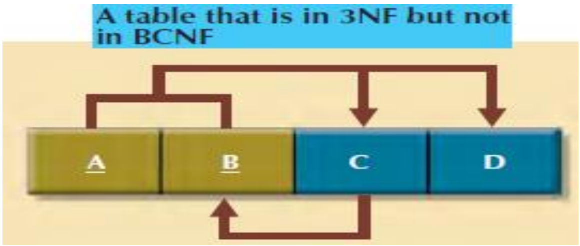
Assign_Hours_Billed. However the derived attribute Assign_Charge in the Assignment table makes easy to write reports or invoices.

Boyce code Normal Form(BCNF):

The BCNF can be violated only when the table contains more than one candidate key

Candidate key:

A key is said to be candidate key if the superkey that does not contain a subset of attributes i.e the key itself a superkey.


A table that is in 3NF but not in BCNF

These functional dependencies are shown below.

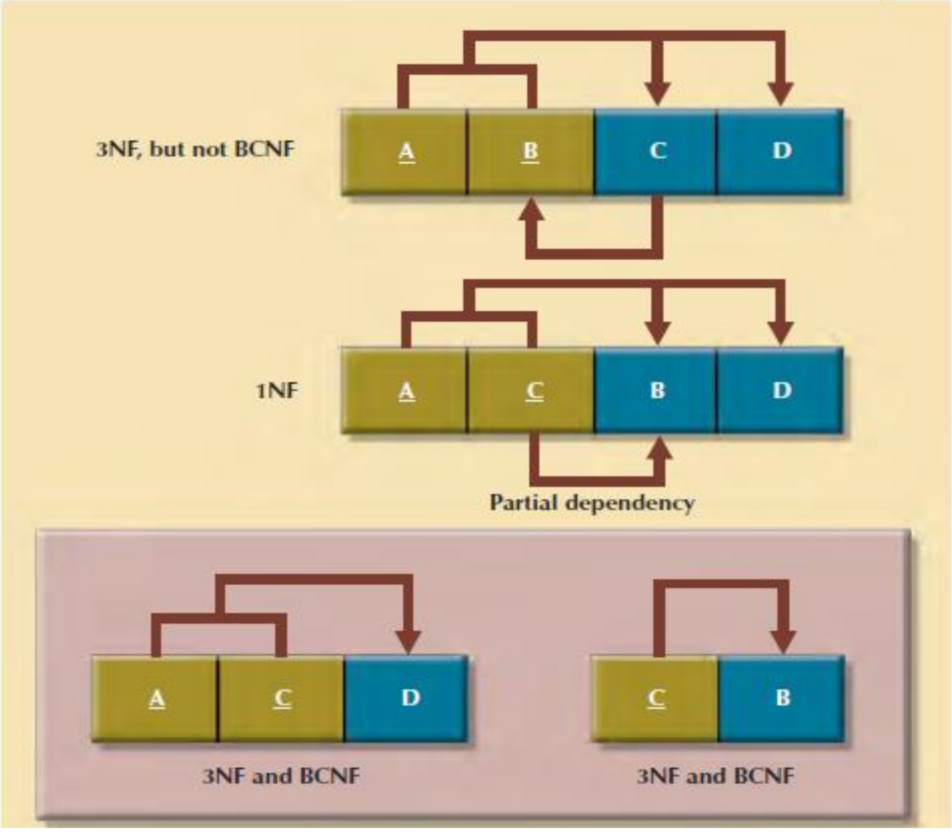
A+B -------------→C,D

C ----------→ B

The table structure has no partial dependency and there is no transitive dependency. But the condition C ---→B indicates that a nonkey attribute determines the part of key the primary key, causes the table to fail to meet the BCNF requirements.

To convert the above table structure from 3NF to BCNF, first change the primary key to A+C. The dependency C---→B means that C is in effect a superset of B. The Decomposition procedures to produce the results shown below.

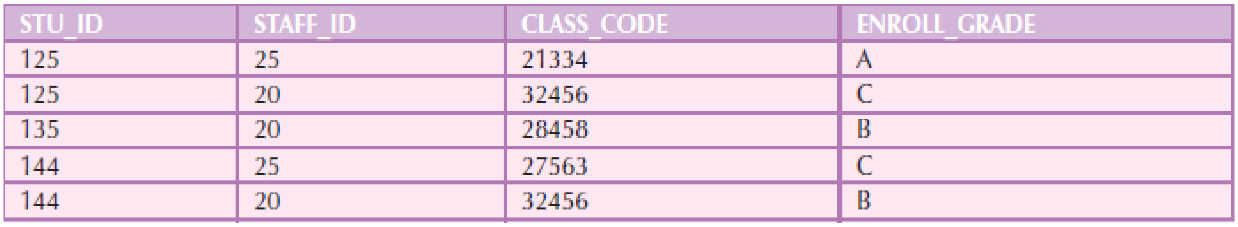
Decomposition to BCNF



Example for BCNF:

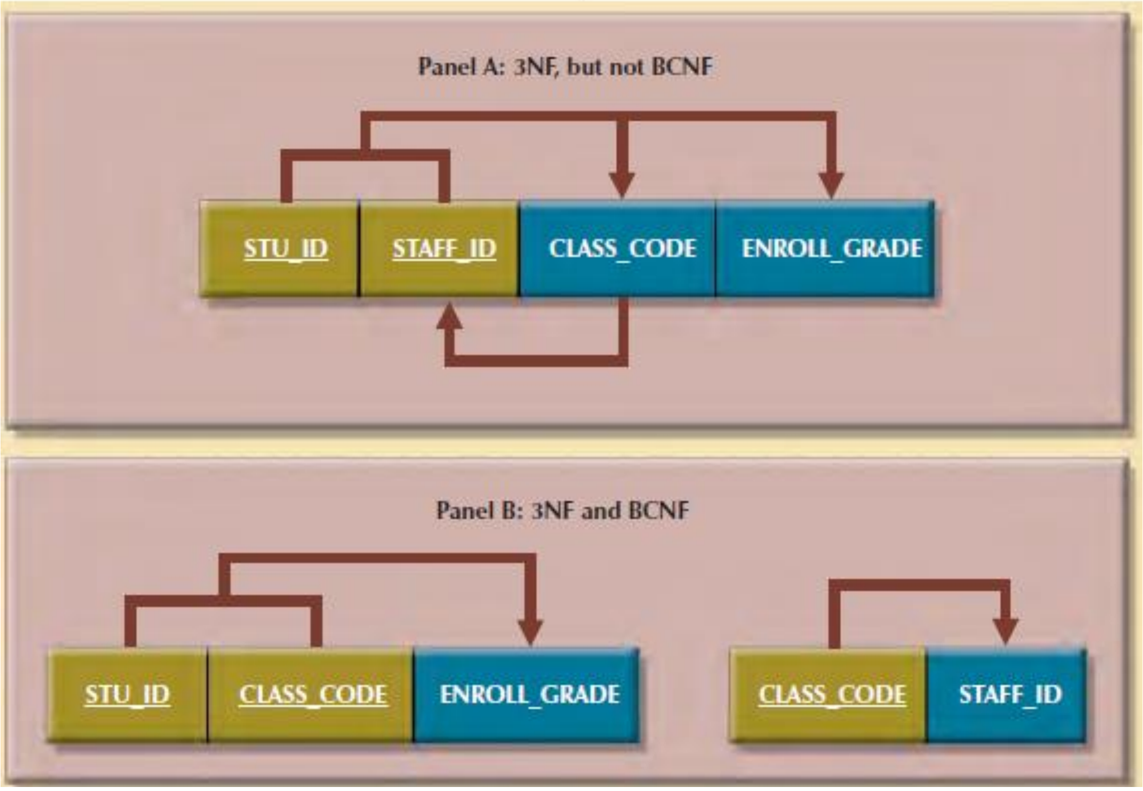
Each Class_Code identifies a class iniquely. A student contains many classes and earning the grades respectively. A staff member can teach many classes. But each class is taught by only one staff member.

Sample Data for a BCNF Conversion



STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

BCNF Decomposition



Stu_Id+Staff_Id --------→Class_Code, Enroll_grade

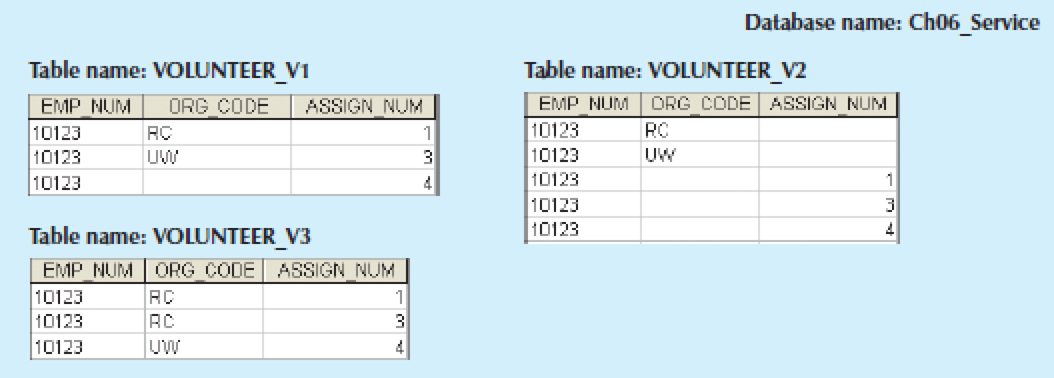
Class_Code ---------------→Staff_Id

The above table contains two candidate keys to violets the BCNF. Now we can eliminate the one candidate key from the above table.

Fourth Normal Form (4NF):

Consider an employee can have multiple assignment i.e, that employee works as an volunteer in service organization and worked in different projects which is shown below

Tables with multivalued dependencies



Database name: Ch06_Service

Table name: VOLUNTEER_V1

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	UW	3
10123		4

Table name: VOLUNTEER_V2

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	
10123	UW	
10123		1
10123		3
10123		4

Table name: VOLUNTEER_V3

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	RC	3
10123	UW	4

The above contains two sets of independent multi valued dependencies(i.e, Org_Code,Proj_Code).If volunteer-1 and Volunteer_2 two tables are implemented. The two tables contains null values.

In volunteer-3 table has a primary key but it is composed of all attributes of the table. When you consider like this it produces many redundancies.

To eliminate multi valued dependency by creating the assignment and service tables as shown below.

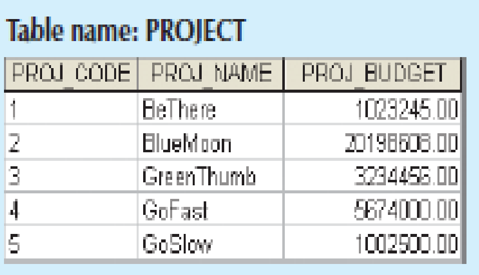


Table name: PROJECT

PROJ_CODE	PROJ_NAME	PROJ_BUDGET
1	BeThere	1023245.00
2	BlueMoon	20198608.00
3	GreenThumb	3234456.00
4	GoFast	5674000.00
5	GoSlow	1002500.00

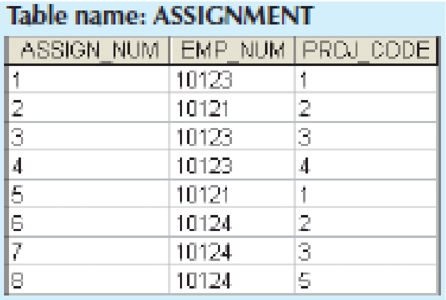


Table name: ASSIGNMENT

ASSIGN_NUM	EMP_NUM	PROJ_CODE
1	10123	1
2	10121	2
3	10123	3
4	10123	4
5	10121	1
6	10124	2
7	10124	3
8	10124	5

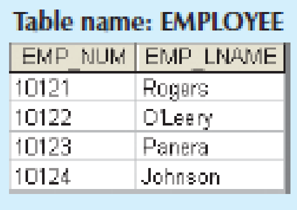


Table name: EMPLOYEE

EMP_NUM	EMP_LNAME
10121	Rogers
10122	O'Leery
10123	Panera
10124	Johnson

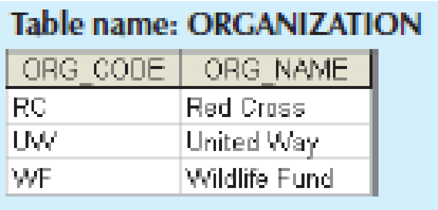


Table name: ORGANIZATION

ORG_CODE	ORG_NAME
RC	Red Cross
UW	United Way
WF	Wildlife Fund

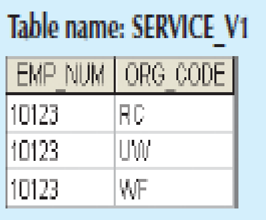


Table name: SERVICE_V1

EMP_NUM	ORG_CODE
10123	RC
10123	UW
10123	WF

In the Assignment table and service table does not contain multi valued dependency

De Normalization: (Under Construction..Ravindra)

Normalization is a very important in database design. Generally the higher normal forms the more the relational join operations required to produce a specific output. A successful design must also consider end user requirement for fast performance. Therefore occasionally we expected to de normalize some portions of the database design in order to meet performance require.

De Normalization produces lower normal forms 3Nf will be converted to 2NF or 2Nf will be converted to 1NF.

Eg: The need for de normalization due to generate evaluation of faculty report in which each row list the scores of obtaining during the last 4 semester taught.

Faculty Evaluation Report:

instructor	**Dept.**	**Sem-1**	**Mean**	**Sem-2**	**Mean**	**Sem-3**	**Mean**	**Sem-4**	**Mean**	**Last_sem avg**

We can generate easy above the report but the problem arises. The data are stored in a normalized table. In which each row represented a different score for a given faculty in a given semester.

EVALDATA:

ID	Instructor	DEPT.	Mean	Semistor

It is some difficulty to generate faculty evaluation report the normalized table

The other table FACHLST faculty history table contains the last four semester mean for each faculty .The faculty history table is a temporary table created from the evaldata as shown below.

Instructor	**Dept.**	**Sem-1**	**Mean**	**Sem-2**	**Mean**	**Sem-3**	**Mean**	**Sem-4**	**Mean**	**Last_sem avg**

The FACHIST is a un normalized from table using the table we can generate .The faculty evaluation report very firstly. After generating the report, the temporary table, FACHIST will be deleted. We are doing like this, we can increase the performance of the database