Normalization

First Normal Form (1st NF)

- The table cells must be of single value.
- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

Do not use multiple fields in a single table to store similar data. For example, to track an inventory item that may come from two possible sources, an inventory record may contain fields for Vendor Code 1 and Vendor Code 2. But what happens when you a dd a third vendor? Adding a field is not the answer; it requires program and table modifications and does not smoothly accommodate a dynamic number of vendors. Instead, place all vendor information in a separate table called Vendors, then link inventory t o vendors with an item number key, or vendors to inventory with a vendor code key.

Second Normal Form (2nd NF)

If it's in 1st NF and if the Primary Key is composite (multiple columns) then any fact in that table must be a fact about the entire composite Primary Key not just part of the Primary Key.

> For example, if an inventory table has a primary key made up of two attributes PartId and WarehouseId. Suppose the inventory table has the warehouse address in it, since warehouse address is a fact about WarehoseId and **not** about the PartId the w arehouse address is in the wrong table. This is in violation of the 2nd Normal Form.

Third Normal Form (3rd NF)

- If it's in the 2nd NF and there are ***no*** non-key fields that depend on attributes in the table other than the Primary Key.

Suppose in the Student table you had student birth date as an attribute and you also had student's age. Students age depends on the student's birth date (a fact about his/her birth date) so 3rd Normal Form is violated. Also, a student table that contains

the address of the Department of his/her major. That address is in the wrong table, because that address should be in the Department table. That address is not a fact about the Student Primary Key. A violation of 3rd Normal Form.

A (non-key) fact in a table should be about the key, the whole key, and nothing but the key.

EXCEPTION: Adhering to the third normal form, while theoretically desirable, is not always practical. If you have a Customers table and you want to eliminate all possible interfield dependencies, you must create separate tables for cities, ZIP code s, sales representatives, and any other factor that may be duplicated in multiple records. In theory, normalization is worth pursing; however, many small tables may degrade performance or exceed open file and memory capacities. But the worst thing is to have to join too many tables in queries. Joining tables is the most expensive operation in time and memory cost.

It may be more feasible to apply third normal form only to data that changes frequently. If some dependent fields remain, design your application to require the user to verify all related fields when any one is changed.

Normalization Examples:

UN-normalized table: Students

Student#	Advisor	Adv-Room	Class1	Class2	Class3
1022	Jones	H412	101-2	112-01	155-01

First Normal Form: NO REPEATING GROUPS

Tables should have only two dimensions. Since one student has several classes, these classes should be listed in a separate table. Fields Class1, Class2, & Class3 in the above record are indications of design trouble.

The next table gets rid of repeating group, class but it is now in violation of the 2nd NF (Normal Form) since Student# is no longer the Primary Key (it now repeats in the table) but Stdent# and Class# can be the Primary Key. But now Advisor is a fact about the Stdent# and **not** a fact about Class#, a violation of

the 2ndNF.

```
Student#   Advisor   Adv-Room    Class#
     ------------------------------------
     1022      Jones      412       101-02
     1022      Jones      412       112-01
     1022      Jones      412       155-01
     4123      Smith      216       201-01
```

We can break up the Students table into two tables Students and Registration

```
    Students:   Student#    Advisor   Adv-Room
                ----------------------------
                1022        Jones       412
                4123        Smith       216

    Registration:   Student#    Class#
                    ------------------
                    1022        101-02
                    1022        112-01
                    1022        155-01
                    4123        201-01
                    4123        211-02
                    4123        214-01
This would be OK but the Adv-Room is a fact about the Advisor and
not the Student a violation of the 3rd NF.
```

Third Normal Form: ELIMINATE DATA NOT DEPENDENT ON KEY

In the last example, Adv-Room (the advisor's office number) is functionally dependent on the Advisor attribute. The solution is to move that attribute from the Students table to the Faculty table, the normalized tables are below:

```
    Students:   Student#    Advisor
                -------------------
                1022        Jones
                4123        Smith

    Faculty:    Name    Room    Dept
                --------------------
                Jones   412     42
                Smith   216     42
Registration:   Student#    Class#
                    ------------------
                    1022        101-02
                    1022        112-01
                    1022        155-01
                    4123        201-01
                    4123        211-02
```

Other Normalization Forms

The **Boyce Codd Normal Form (BCNF**) which is a refinement of the 3rd Normal Form, the 4th Normal Form and higher. In practice people rarely go beyond the 3NF and almost never beyond the 4NF.

Q10 What is the **BCNF**? I heard of it, but I did not really understand it.

Q10. An example may help. Suppose a student can have more than one major and we would like to keep track of the student's major(s) and students advisors in the following table:

Students	**StudentId**	**Major**	**Advisor**
	100	Math	Hilbert
	150	Psychology	Jung
	200	Math	Courant
	300	Psychology	Ruth
	300	Com. Sci.	Vasilaky

Since StudentId repeats it can't be the Primary Key. We can choose either StudentId and Major as the Primary Key or StudentId and Advisor as Primary Key. Say we choose StudentId and Major as the Primary Key. But that means that the remaining field Advisor is a fact about both StudentId and Advisor. So we know that Hilbert is an advisor for math majors and advises student 100.

This table is in 3NF but it still has anomalies (inconstancies). It's in 2NF because the advisor is a fact about both the student advised and the major he/she advises. It's in 3Nf because advisor is a fact only about the primary key (StudentId, Major).

Suppose student 300 drops out of school. If we delete student 300 we lose the fact that Dr. Ruth Advises psychology. This is a deletion anomaly. Also how can we know that Dr. Freedman advises Economics until student major in Economics? This is an insertion anomaly. So we have inconsistent dependency.

An attribute is a determinant if it determines another attribute. For example StudentId determines Major. Advisor determines the major she/he advises.

BCNF

A table is in BCNF if it's in 3rd NF and every determinant can be used as a Primary Key.

In our example Advisor attribute determines Major but is not a possible Primary Key. StudentId and Advisor together is a possible (candidate) Primary Key.

Normalized:

STUDENTS

StudentId	Advisor
100	Hilbert
150	Jung
200	Courant
300	Ruth
300	Vasilaky

ADVISORS

Advisor	Major
Courant	Math
Vasilaky	Comp Sci
Ruth	Psychology
Hibert	Math
Jung	Psychology

Fourth Normal Form A table is in the 4NF if it's in BCNF and has no attribute with mutivalued dependencies.

Suppose a Student can have more than one major and more than one activity. For example:

StudentId	Major	Activity
100	Music	Swimming
100	Accounting	Swimming
100	Music	Tennis
100	Accounting	Tennis
150	Math	Jogging

Note that all three attributes make up the Primary Key.

Note that StudentId can be associated with many major as well as many activities (multivalued dependency). Multivalued dependency lead to modification anomalies. Suppose student 100 signs up for skiing. Then we would insert (100, Music, Skiing)

This row implies that student 100 skies as Music major but not as an accounting major, so in order to keep the data consistent we must add one more row (100, Accounting, Skiing). This is an insertion anomaly. Here are the tables Normalized:

Student-Major

StudentId	**Major**
100	Music
100	Accounting
100	Math

Student-Activity

StudentId	Activity
100	Skiing
100	Swimming
100	Tennis
150	Jogging