NORMALIZATION

The process of normalization is related to the theory of logical database design. Normalization theory is built around the concept of NORMAL FORMS.

Normalization is the process of transforming some objects into a structural form that satisfies some collection of rules. Any schema that is in normal form is guaranteed to have certain quality characteristics.

Each normal form has a rule that describes what kinds of functional dependencies the normal form allows.

- Normalization is the process of transforming schemas in order to remove violations of the normal form rules.
- Normalization is applied independently to each relation schema in a database schema.
- A database schema is said to be *in normal form* if each of its relation schemas is in the normal form.

Problems with bad-designed (not normalized relational databases) can be detected without any other analysis tool can be stated are as follows:

1. Anomalies that imply additional work to be done during insertion into and modification of a relation, and that may cause accidental loss of information during a deletion from a relation.
2. Waste of storage space due to nulls and difficulty of performing aggregation operations and joins due to null values.
3. Generation of invalid and spurious data during joins on improperly related base relations.

The general criteria for quality of a database design:

- Each attribute and schema should have a simple meaning.
- Redundant values in tables should be minimized.
- The presence of null values in tables should be minimized.
- Spurious or meaningless entries should not be allowed.

Benefits of Data Normalization:

The major benefits of correctly normalized database from the MIS generally include:
. Development of strategy for constructing relations and selecting key.
. Improved interfaces with end-user computing activities.
. Reduced problems associated with inserting and deleting data.
. Reduced enhancement and modification time associated with the data structure.
. Improved information for decisions relating to physical database design.
. Identification of potential problems that may require additional analysis & documentation.
. Minimizing redundancy
. Minimizing the insertion, deletion, and update anomalies.

From the end-users perspective, a correctly normalized database will translate into improved response time from the MIS organization as well as improved capabilities for end user computing activities.

We have 3 basic, and some more additional normal forms. First three forms based on the ***functional dependencies***. Fourth one depends on the ***multi-valued dependency*** and the fifth one depends on the ***join dependency***

Key attribute is an attribute that is part of a key (either primary key or candidate key)

Non-key attribute is an attribute that is not part of any key

Primary key is one of the keys that has been selected to identify the objects of the schema

FIRST NORMAL FORM
(Functional dependency)

It states that the domain of an attribute must include **only atomic (simple, indivisible) values** and that the value of any attribute in a tuple must be single value from the domain of that attribute.

1NF disallows "relations within relations" or "relations as attributes of tuples".

Functional dependency is a strong connection between two or more attributes within a table.

Given a relation R, an attribute y is **functionally dependent** on another attribute x (x generally will be the primary key) if and only if each x-value in R has associated with exactly on y-value. (x functionally determines y)

Example- 1) The following relation is NOT in the first normal form. Find and show the Functional Dependencies :

EMPLOYEE (EMPNAME, EADRS, EMPNO, DEPTNO, DNAME, DEPTHEAD, JOB)

EMPLOYEE (EMPNAME, EADRS, EMPNO, DEPTNO, DNAME, DEPTHEAD, JOB)

In the above example, if you know the employee no, you can define the name and address of the employee. So employee name functionally dependent on EMPNO. Other functional dependencies can also be explained in the same manner. And it is given as an information that, every employee may have different jobs in different departments.

A key constraint is a functional dependency.

Solution:

EMPLOYEE (EMPNAME, EADRS, EMPNO)

DEPARTMENT (DEPTNO, DNAME, DEPTHEAD)

JOBS (EMPNO, DEPTNO, JOB)

Example-2)

The following base relation is given. Find and show the Functional Dependencies :

CHARGES(PNO, PNAME, ENO, ENAME, JOB_CLASS, CHG_HOUR, HOURS)

CHARGES(PNO , P-NAME , E-NO , E-NAME , JOB-CLASS , CHG-HOUR , HOURS)

Charges table contains a primary key composed of the attributes P-NO and E-NO. Any attribute that is at least part of a key is known as a prime attribute or a key attribute. Therefore, both P-NO and E-NO are prime attributes.

Solution:

PROJECT (PNO, P-NAME)

EMPLOYEE (E-NO, E-NAME, JOB-CLASS, CHG-HOUR)

ASSIGNMENTS (PNO, E-NO, HOURS)

1 NF Definition
The term ***first normal form (1NF)*** describes the tabular format in which 1. All the key attributes are defined. 2.There are no repeating groups in the table. In other words, each row/column intersection can contain one and only one value, rather than a set of values. (atomic values) All multivalued attributes removed. 3. All attributes are dependent on the primary key. (functional dependency-key constraint)

SECOND NORMAL FORM

(Full functional dependency)

A relation is said to be in second normal form if it is in 1NF and one of the following condition applies:

1. Key contains of a single attribute.
2. No non-key attribute exists.
3. Every non-key attribute fully functionally depends on the entire key, which is composite.

===> ***Every nonprime attribute A in R is not partially dependent on any key or R.***

Example: Show the functional dependencies

GRADES(<u>STUID, COURSE_CODE</u>, STU_NAME, LETTER_GRADE)

PARTIAL DEPENDENCY

GRADES(STUID, COURSE_CODE, STU_NAME, LETTER_GRADE)

In this relation STU_NAME can be identified by only STUID. There is no need to have COURSE_CODE as a part of the primary key for identification of name. So there is a **PARTIAL DEPENDENCY** between composite primary key and STU_NAME non key attribute.

Solution:

GRADES(STUID, COURSE_CODE, LETTER_GRADE)

FULL FUNCTIONAL DEPENDENCY

STUDENT(STUID, STU_NAME)

2 NF Definition
A table is in 2NF if 1. It is in 1NF and 2. It includes no partial dependencies; that is , no attribute is dependent on only a portion of the composite key. All partial dependencies removed.

THIRD NORMAL FORM

(No transitive dependency (non key --> non key)

If a relation is in second NF and if there is no ***transitive dependency*** among its attributes, the relation is said to be in third normal form.

====> ***Every nonprime attributes :***

**** Fully functionally dependent on every key of R,***

**** Non transitivity dependent on every key of R.***

Example-1) Define missing Functional dependencies

EMPDEPTS (ENAME, <u>SSN</u>, BDATE, ADR, DNUMBER, DNAME, MGR)

EMPDEPTS (ENAME, SSN, BDATE, ADR, DNUMBER, DNAME, MGR) NOT IN 3NF(since p.k. is given, it will be accepted as in 1NF.)

Transitive dependency

Solution:

EMPLOYEE (ENAME, SSN, ADR, DNUM)

DEPARTMENT(DNUMBER, DNAME, MGR)

Example-2)

There is transitive dependency between JOB-CLASS and CHG-HOUR if chg-hour is changing for every job-class.

PROJECT (PNO, P-NAME)

Transitive dependency

EMPLOYEE (E-NO, E-NAME, JOB-CLASS, CHG-HOUR)

ASSIGNMENTS (PNO, E-NO, HOURS)

Solution:

PROJECT(P-NO, P-NAME)

ASSIGN(P-NO,E-NO, HOURS)

EMPLOYEE(E-NO, E-NAME, JOB-CLASS)

JOB(JOB-CLASS, CHG-HOUR)

3 NF Definition
A table is in 3NF if 1. It is in 2NF and 2. It contains no transitive dependencies. All transitive dependencies removed. Non of the non_key attributes functionally identifies any other non_key attribute.

THE BOYCE-CODD NORMAL FORM (BCNF)

(Kind of transitive dependency (non key --> part of the key)

If every determinant of relation is a candidate key, this relation is in BCNF form.

Example-1)

transitive dependency from non-key to part of the key

AP-WHOLESALER (CENTER, PART, EMP-NUM, IN-STOCK, STATUS)

SEA........BE11....147............23 ....OK

SEA........AL34.....345............34..............OVERSTOCK

ALT.........BE55....268............12..............OK

SEA........BB67.....147...........15..............OK

In this example, center and part constitutes the primary key of the relation. Other attributes are functionally depends on the complete primary key values. And there is not exists any transitive dependency. But, EMP-NUM functionally defines the CENTER, since an employee always work on the same center. Giving the identification of an employee can define the center of this process. This is the special case of the 3NF. In that case the new database should be like this:

EMPLOYEE (EMP-NUM, CENTER)

AP-WHOLESALER (EMP-NUM,PART, IN-STOCK, STATUS)

Example-2)

In the high school, all the *teachers assigned to only one type of course*. For one course, there can be more than one teacher, but for one teacher there will be single course!

TEACH (STUDENT , COURSE , INSTRUCTOR)

Functional dependency 1: {STUDENT, COURSE} → INSTRUCTOR
Functional dependency2: INSTRUCTOR → COURSE

It is clear that functional dependencies in the first normal form is not defined correctly. But, since primary key is defined, we have to analyze and decompose in the BCNF. So the decomposition will be like this:

TEACHER_COURSE(INSTRUCTOR, COURSE)

STUDENT_INST(INSTRUCTOR, STUDENT)

Example-3)
If the course is given by only one teacher:

ENROLL(STUD,TEACHERID,COURSE_CODE, GRADE) **NOT fit into Boyce-Code NF.**

ENROLL(STUID,COURSE-CODE,GRADE)

OFFER(COURSE-CODE, TEACHERID)

Normalized ones

Boyce-Codd Normal Form Definition
A table is in BCNF if every determinant in that table is a candidate key. If table contains only one candidate key, 3NF and BCNF are equivalent. Non of the nonkey attribute functionally identifies a part of the composite primary key. There is not any transitive dependency between nonkey attributes and part f the keys.

FOURTH NORMAL FORM

(No multiple sets of multivalued dependency)

If a relation is in BCNF and no Multi Valued Dependency (MVD) exist, the relation said to be in 4NF.

MVD: If there is more than one opportunity on attribute x, and the value of x dependent the value of y, we can say that there is a MVD of x on another attribute y.

Example:

EMPLOYEE(ENAME, PNAME, DNAME)

ENAME	**PNAME**	**DNAME**
Smith	Red Cross	John
Smith	Red Cross	Anna
Smith	United Way	John
Smith	United Way	Anna

ENAME	**PNAME**	**DNAME**
Smith	Red Cross	
Smith	United Way	
Smith		John
Smith		Anna

In such cases, although the table is in BCNF, the occurrence of the tuples are meaningless (MVD). So the relation should change into:

EMP-PROJ (<u>ENAME, PNAME</u>)
EMP-DEPT (<u>ENAME, DNAME</u>)

ENAME	PNAME
Smith	Red Cross
Smith	United Way

ENAME	DNAME
Smith	John
Smith	Anna

4 NF Definition
A table is in 4NF 1. if it is in 3NF and 2. has no multiple sets of multi-valued dependencies.

FIFTH NORMAL FORM

If every join dependency in R is implied by the candidate keys of R, we can say that R is in 5NF. Some authors calls this form as Project Join Normal Form. (PJNF)

EXAMPLE:

SNAME	**PARTNAME**	**PROJNAME**
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

Each supplier may supplies different parts and different parts may be used in one project. So suppliers supplies to different projects.

Whenever supplier supplies part p, and a project j uses part p, and the supplier s supplies at least one part to project j, then supplier s will also be supplying part p to project j. This constraint can be restated in other ways and specifies a join dependency JD(R1, R2, R3) among the three projections
R1(SNAME, PARTNAME),
R2(SNAME, PROJNAME),
R3(PARTNAME, PROJNAME) of SUPPLY.

LOTS EXAMPLE FOR 1NF, 2NF, 3NF, BCNF:

These facts are given beforehand for the below example:

1. Every country has its own tax_rate
2. Prices are the predefined minimum price per square meter of the area.

LOTS (propid, country_name, lot#, area, price, tax_rate)

Candidate key

1NF/ not in 2NF

1NF / 2NF

Partial dependency

LOTS1(propid, country_name, lot#, area, price) 2NF / not in 3NF

Transitive dependency

LOTS2(country_name, tax_rate) 2NF /3NF

LOTS3(area, price) 3NF /BCNF

LOTS1(propid, country_name, lot#, area) 3NF / Not in BCNF

LOTS4(area, price, country_name) BCNF

LOTS5(propid, area, lot#) BCNF

SUMMARY:

1NF: Define functional dependencies by eliminating repeating values and decompose given relation into relations which are satisfying key constraints.

2NF : R1($\underline{A, B}$, C, D, E) ==> R1($\underline{A, B}$, C , E)
R2($\underline{A}$, C)

3NF: R1($\underline{A}$, B, C, D , E , F) ==> R1($\underline{A}$, B, C , D , E)
R2($\underline{D}$, F)

BOYCE CODE: R1 ($\underline{A, B}$, C, D, E) ==> R1($\underline{A, D}$, C E)
R2($\underline{D}$, B)

4NF: R1(A, B, C) in which, A has multivalues relation with B and C,

==> R1(A , B)
R2(A, C)

5NF: R1(A, B, C) in which, A has multivalued dependency with B and C, AND B has multivalued dependency with C

==> R1(A, B)
R2(A, C)
R3(B, C)