Lesson 18

Objectives

- Data Normalization
  - Characteristics of Suitable Relation
  - Advantages of Suitable Relations
  - Goal of Normalization
  - Data Anomalies
    - Insertion Anomaly
    - Deletion Anomaly
    - Updation Anomaly
  - Definitions of Some Terms
    - Functional Dependency
    - Partial Dependency
    - Transitive Dependency
  - First Normal Form
  - Second Normal Form
  - Third Normal Form

Data Normalization

- It is a technique to validate and improper logical design. So that it satisfy certain constrain that avoid unnecessary duplication of data
- The process of decomposing relation with anomalies to produce smaller well structured relations
- It is a technique for producing a set of suitable relations that support the data requirements of an enterprise

Characteristics of Suitable Relation

- Minimal no. of attributes necessary to support the data requirements
- Attribute with a close logical relationship found in the same relation
- Minimal redundancy with each attribute represented only once with important expectation of attribute that from all or part of foreign key. In other words we can say that redundancy will never obsolete but will be minimized. For example foreign key will be repeated. So there should be exception for foreign key to repeat itself.

Advantages of Normalization

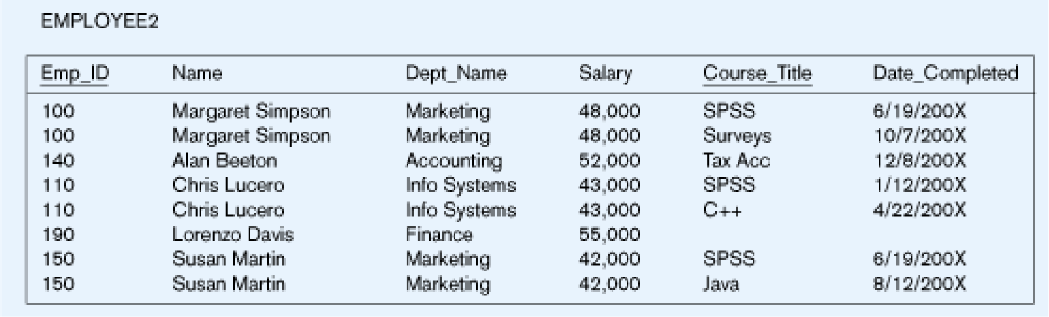
- Maintenance will be easier
- Access will be easier
- Minimal storage

Goal of Normalization

Main goal of normalization is to avoid anomalies.

Insertion Anomaly-adding new rows forces user to create duplicate data.
Insertion Anomaly-deleting rows may cause a loss of data that would be needed for other future rows.
Insertion Anomaly-changing data in a row forces changes to other rows because of duplication.

Example



EMPLOYEE2

Emp_ID	Name	Dept_Name	Salary	Course_Title	Date_Completed
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/200X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/200X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/200X
110	Chris Lucero	Info Systems	43,000	SPSS	1/12/200X
110	Chris Lucero	Info Systems	43,000	C++	4/22/200X
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/19/200X
150	Susan Martin	Marketing	42,000	Java	8/12/200X

In the above relation all three anomalies exist as:

- **Insertion** – can't enter a new employee without having the employee take a class. In other words we only wants to add data of employee but we are forced to enter data of class (course_title, date_completed).
- **Deletion** – if we remove employee 140, we lose information about the existence of a Tax Acc class because there is only one employee (having Id 140) who is enrolled in Tax Acc class. So the deletion of 140 will also lead to delete Tax Acc. Due to this information about Tax will also be deleted.
- **Modification** – giving a salary increase to employee 100 forces us to update multiple records because 100 employee exist in multiple records.

Definitions of Terms
Functional dependency

If value of one attribute (A) determine the value of another attribute B) then we say that B is functional dependent on A. Each non key field is functional dependent on every candidate key.
Functional dependency between A and B can be shown as

$A \xrightarrow{\text{B is functionally dependant on A}} B$

Example

Assume a relation branch(bno, branchAddress). Here we can determine branchAddress from bno and vice versa. Hence both are functional dependant on eatch other.

branchAddress ⟶ bno

bno ⟶ branchAddress

Another example of Invoice relation as
Invoice(orderNo, orderDate, customerID, customerName, customerAdress, productID, productPrice, productDescription, orderQuantity).

In invoice table following functional dependencies exist.

orderNo → orderDate
orderNo → customerID
orderNo → customerName
orderNo → customerAdress
orderNo → orderQuantity
customerID → customerName
customerID → customerAdress
productID → productPrice
productID → productDescription

There is no single attribute that has an ability to determine values of the remaining attribute. From orderNo we cannot determine the value of productID uniquely. Therefore a composite key is made by combining orderNo and productID. This combination has capability to determine the values of remaining attributes uniquely. In other words we can say that all of the remaining attributes are functionally dependent on orderNo and productID. So

Partial Dependency

If value of one or more non key attributes can be determine by part of composite key, then it is said that attributes that are determined by part of key are partially dependent on primary key.

Example

Invoice(orderNo, orderDate, customerID, customerName, customerAdress, productID, productPrice, productDescription, orderQuantity).

In the above relation there is a composite key (productID&orderNo); orderDate, customerID, customerName, customerAdress can be determined by using only orderNo that is a part of composite key.
In this scenario; orderDate, customerID, customerName, customerAdress are partially dependent on primary key. Therefore we say that there is a partial dependency in relation Invoice.

If there is no composite key in a relation then there will be no partial dependency in a relation.

Transitive dependency:-

In a relation; if one non key attribute is functionally dependent on another non key attribute then we will say that there is a transitive dependency exists in that relation.

Example

Invoice(orderNo, orderDate, customerID, customerName, customerAdress, productID, productPrice, productDescription, orderQuantity).

In this example customerName and customerAdress are functionally dependent on customerID. So transitive dependency exist in Invoice relation

UN _Normalized form:-

A table that contains one or more repeating groups (multivalued attributes)

To create UN _normalized form we take following steps

➔ Transform the data from the information resources into table format with columns and rows
For this purpose we do.
1-> First normal form (1NF)
2-> Second normal form (2 NF)
3-> Third normal form (3 NF)

FIRST NORMAIL FORM (1 NF)

A table in which the intersection of each row and columns contains one and only one value. Means each cell contains atomic value.

If a table qualifies as a relation then it will be in first normal form. In other words; each relation will must be in 1NF.

UNF TO 1NF

1. Nominate an attribute or group of attributes to act as the key for the un-normalized table.
2. Identify the repeating group in the un-normalized table which repeats for the key attribute.
3. Remove the repeating group by
    a. Entering appropriate data into the empty column of rows containing the repeating group. OR
    b. Placing the data of repeating group along with a copy of the original key into a separate relation

SECOND NORMAIL FORM (2 NF)

A relation will be in second normal form if it is in (1 NF) and there is no partial dependency in that relation. OR

A relation that is in 1NF and every non key attribute is fully functional dependent on the key.

1NF TO 2NF

- Identify the primary key in the 1NF relation.
- Identify the functional dependencies in the relation.
- If partial dependency exists on the primary key then remove them by placing them in new relation along with a copy of their determinants.

Third Normal form (3 NF)

A relation that is in (2NF) and in which there is no transitive dependency exists will be in 3NF.

2NF to 3NF

Indentify the primary key in the 2NF relation.

Identify functional dependencies in the relation.

If transitive dependencies exist on the primary key; remove them by placing them in a new relation along with a copy of their determinant.