DATABASE - 1

3RD CLASS

COMPUTER SCIENCE DEPARTMENT

8th Lecture – Merging Relations

Sunday 10th of November 2024

LECTURER :

DR. RAYAN YOUSIF ALKHAYAT

MERGING RELATIONS:

- As a part of logical design process , normalized relations may have been created from number of separated E-R Diagrams and (possibly) other users' views , some of the relations may redundant as ; that is ,they may refer to the same entities .
- If so, we should merge those relations to remove redundancy. This section describes merging relations also called (*view integration*).

- An understanding of how merge relations is important for three reasons :

1. On large projects: the work of several sub teams becomes together during logical design , so there is a need to merge relations .
2. Integrating existing databases with new information requirements often leads to the need to integrate different views.
3. New data requirements may arise during life cycle, so there is a need to merge any new relations with what has already been developed.

An example:

Suppose that modeling user view results in the following 3NF relation:

EMPLOYEE1 (Employee_ID, Name , Address, Phone)

Modeling a second user view might result in the following relation:

EMPLOYEE2 (Employee_ID, Name , Address, Jobcode,No_years)

Since this two relations have the same primary key (Employee_ID) ,they likely describe the same entity and may be merged into one relation , the resulting of merging the relations is in the following relation :

EMPLOYEE(Employee_ID, Name , Address, Phone, Jobcode,No_years)

Notice that the attributes in both relations (such as Name in this example) appear only once in the merged relations.

View Integration Problems:

- When integration relations as in the preceding example ,the database analyst must understand the meaning of the data must be prepared resolve any problems that may arise in that process.
- In this section we describe a briefly illustrate for problems that arise in view integration synonyms , homonyms, transitive dependences and supertype/subtype relationship,

1. Synonyms

- in some situations, two(or more) attributes may have different names but the same meaning ,as when they describe the same characteristics of an entity .
- Such attributes called synonyms, for example Employee_ID and Employee_Number may be synonyms.
- When merging relations that contains synonyms you should obtain agreement of (if possible) from user on a single, standardized name for the attribute and eliminate other synonyms .
- Another alternative is to chose a **third name** to address the synonyms.

for example, consider the following two relations :

STUDENT1 (Student_ID,Name)

STUDENT2 (Matriculation_Number ,Name, Address)

In this case the analyst realized that both Student_ID and Matriculation_Number are synonyms for Social Security Number (SSN) are identical attributes .

One possible solution is would be standardized on the two attributes names such as Student_ID .

Another option is to use a new attribute name such as SSN, to replace both synonyms . Assuming the later approach merging the two relations would produce the following results :

STUDENT (SSN, Name, Address)

- Often, when there are synonyms there is a need to allows some database users to refer to the same data by different names , users may need to use familiar names that are consistent with terminology of there part of the organization .
- An Alias is an alternative name used for an attribute .
- Many database management systems allow the definition of alias that may be used interchangeably with primary attribute label

2. Homonyms

an attribute that may have more than one meaning is called a Homonyms , for example the term account might refers to bank's checking account , saving account , loan account , or other type of account, (therefore account may refer to the different data , depending on how it is used) .

You should be lockout for homonyms when merging relations , consider the following example ;

STUDENT1 (Student_ID,Name, Address)

STUDENT2 (Student_ID ,Name, Phone_No, Address)

- In discussions with user the analyst discovered that the attribute Address in STUDEN1 refers to student campus address , while in STUDENT2 the same attribute refers to student permanent address(or home address) to resolve this conflict we would probably need to create new attributes name , so that merged new relation will become :

- STUDENT (Student_ID ,Name, Phone_No, Campus_Address , Permanent_Address)

3. Transitive dependencies

when two 3NF relations are merged to form a single relation , transitive dependency may result . For example, consider the following two relations :

STUDENT1 (Student_ID , Major)

STUDENT2 (Student_ID , Advisor)

Since STUDENT1 and STUDENT2 have the same primary key , the two relations may be merged :

STUDENT (Student_ID , Major, Advisor)

However, suppose that each major has exactly one advisor .

In this case Advisor is functionally dependent on Major.

Major → Advisor

If the preceding functional dependency exist ,the student is in 2NF but not in 3NF since it contains a transitive dependency , the analyst can create 3NF relations by removing the transitive dependency (Major becomes foreign key in STUDENT) ;

STUDENT1 (Student_ID , Major)

MAJOR ADVISOR (Major, Advisor)

4. Supertype/ subtype relationships

these relationships may be hidden in user views or relations . Suppose that we have the following two hospital relations:

PATIENT1 (Patient_ID, Name, Address)

PATIENT2 (Patient_ID, Room_No)

Initially it appears that these two relations can be merged into a single PATIENT relation . However the analyst correctly suspects that there are two different types of patients inpatients and outpatients .

PATIENT1 actually contains attributes that common to all patients.

PATIENT2 contains an attributes (Roon_No) that is a characteristic only for inpatients . In this situation the analyst should create a supertype’ subtype relation ship for these entities :

PATIENT (Patient_ID, Name, Address)

INPATIENT (Patient_ID, Room_No)

OUTPATIENT (Patient_ID, Date_Treated)

END OF LECTURE 8

PRINCIPLES OF DATABASE

2ND CLASS

NETWORKS DEPARTMENT

9th Lecture - Physical Database Design

17-11-2024

LECTURER :

DR. RAYAN YOUSIF ALKHAYAT