

Department of Software Engineering
Mehran University of Engineering and Technology, Jamshoro

Course: SW222 – Database Management & Administration

Instructor	Ms Shafiya Qadeer	Practical/Lab No.	04
Date	03/04-02-2021	CLOs	2
Signature		Assessment Score	2 Marks

Objectives - To become familiar with Data De- Normalization

Lab Discussion: Theoretical concepts and Procedural steps

Database normalization:

- Database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them.
- The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database using the defined relationships.

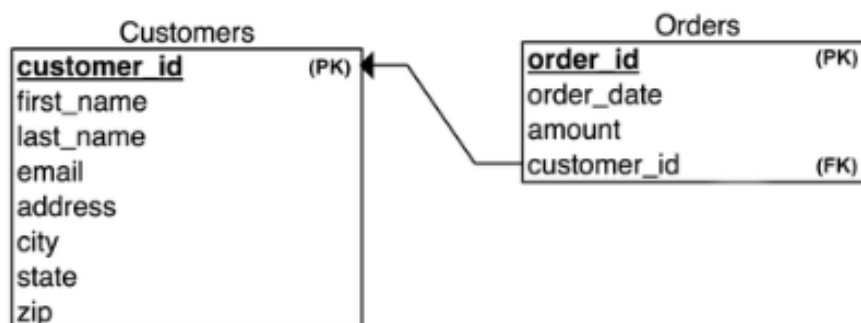
Good for OLTP(Online transaction processing system)

DISADVANTAGES OF NORMALIZATION

1. running time of query increases because of joins.
2. It decreases database performance.

JOINS

- A JOIN clause is used to combine rows from two or more tables.



WHAT IS DENORMALIZATION

- It is the reverse process of Normalization i.e, to combine two or more tables into a single table.
- De-normalization increases the performance (Searching data from one table is quiet faster than searching data from multiple tables).
- Good for OLAP(Online analytical Processing system)

WHY DO WE NEED DENORMALIZATION

- The usual goal is to decrease the running time of select queries by making data more accessible to the queries.
- and to increase the performance of a database.
- Denormalization makes data retrievals easier to express and speeds up the database performance.
- So, denormalize only when there is a very clear advantage to doing it.
- Low number of updates
- Large number of selects and fetching data

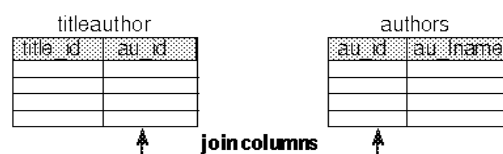
METHOD OF DENORMALIZATION

- 1) Adding Redundant Columns.
- 2) Combining Tables
- 3) Repeating Groups

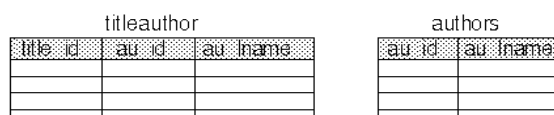
1. Adding Redundant Columns

- You can add redundant columns to eliminate frequent joins. For example, if frequent joins are performed on the titleauthor and authors tables in order to retrieve the author's last name, you can add the au_lname column to title author.

```
select ta.title_id, a.au_id, a.au_lname
from titleauthor ta, authors a
where ta.au_id = a.au_id
```



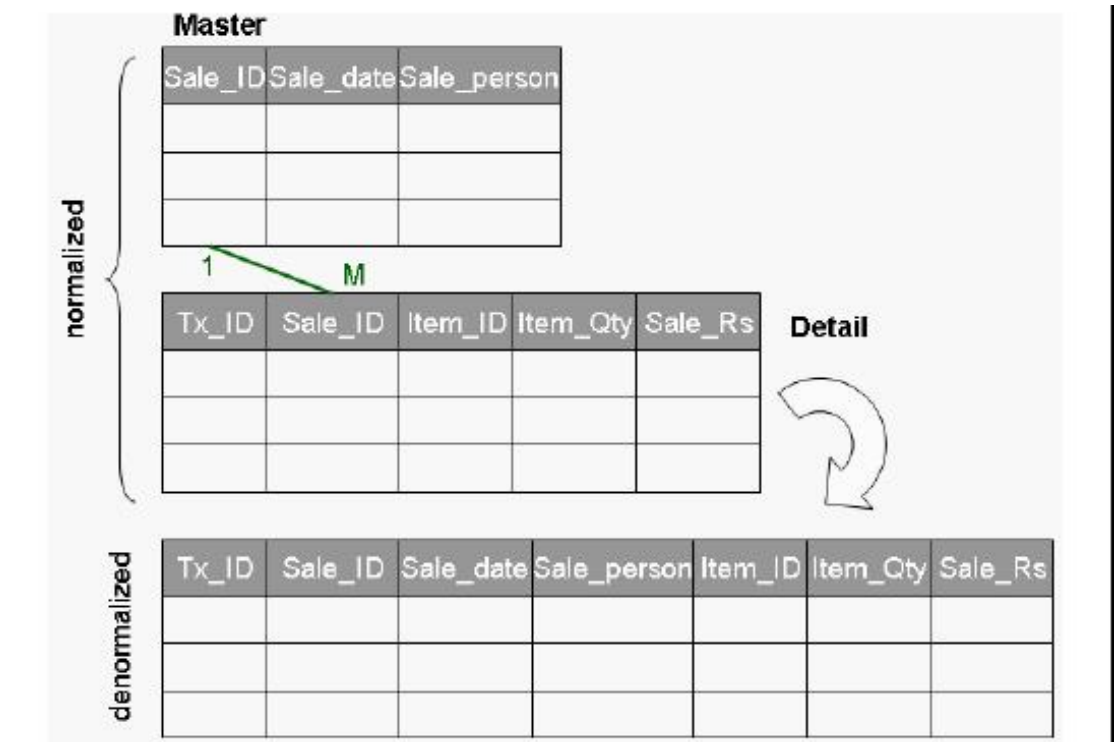
```
select title_id, au_id, au_lname
from titleauthor
```



- Adding redundant columns eliminates joins for many queries. The problems with this solution are that it:
- Requires maintenance of new column. All changes must be made to two tables, and possibly to many rows in one of the tables.
- Requires more disk space, since *au_lname* is duplicated.

2. Combining Tables

- Combine two or more tables into a single table to eliminate the joins and increase the database performance.



3. Repeating Groups

The repeating groups can be stored as a nested table within the original table.

- **example:**(before denormalization)

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Telephone

telNo	branchNo
0207-886-1212	B005
0207-886-1300	B005
0207-886-4100	B005
01224-67125	B007
0141-339-2178	B003
0141-339-4439	B003
0117-916-1170	B004
0208-963-1030	B002

after denormalization

Branch

branchNo	street	city	postcode	telNo1	telNo2	telNo3
B005	22 Deer Rd	London	SW1 4EH	0207-886-1212	0207-886-1300	0207-886-4100
B007	16 Argyll St	Aberdeen	AB2 3SU	01224-67125		
B003	163 Main St	Glasgow	G11 9QX	0141-339-2178	0141-339-4439	
B004	32 Manse Rd	Bristol	BS99 1NZ	0117-916-1170		
B002	56 Clover Dr	London	NW10 6EU	0208-963-1030		

ADVANTAGES

- Minimizing the need for joins.
- Reducing the number of foreign keys in relations.
- Reducing the number of relations/tables.
- **DISADVANTAGES**
- **Updates:** May speed up retrievals but can slow down updates.
- **Disk Space:** need more memory space to store redundant data.
- **Data anomalies:** We have to be very aware of the fact that data now can be changed in more than one place. We must adjust every piece of duplicate data accordingly.

EXAMPLE:(MUET CASE STUDY)

EXAMPLE:(MUET CASE STUDY)

MUET contains many faculties. Each of the faculty is identified by a unique faculty ID number. And has a name.

There are many departments in each of the faculty. Each department has a unique department ID number and a name.

Every teacher has a unique teacher ID, name, father's name, mobile number, CNIC number and is associated with one of the department.

Every student has a unique roll number, name, father's name, mobile number, CNIC number and is associated with one of the department.

A teacher can teach multiple courses during a session.

A student can take multiple courses during the session.

Every course is identified by a unique course ID number and has course name.

- **convert the entitites into tables and attributes in columns of the table.**
- **Below diagram shows that the tables are in normalized form**
- **Perform** Combining Tables method on the tables shown in diagram

FacID	Facname

st_ID	st_name

c_id	c_name

T_ID	T_name

Deptno	dname

To convert it in denormalization, combine all the tables to make it a single table

[illegible]

- Consider the image below. The top contains several distinct tables that encapsulate logically separate bits of info. The bottom shows the results of those tables joined together. This is denormalization.

Lab Tasks

1. Convert Following Data Base into De normalization form

