Individual Coursework

Sameer Tamang

Student ID:210369 / CU ID:12336463

BSc. (Hons.) Ethical Hacking & Cyber Security, Softwarica College of IT and E-commerce, Coventry University

ST4056CEM Introduction to Web Development & Database System

Niman Maharjan

27th July, 2022

Contents:

Task 1 2

Task 2 5

Task 3 6

Task 4 7

Task5 11

Task 6 13

Task 7 14

Reference 16

Task 1

The practice of effectively structuring data in a database is called normalization. The normalization process has two specific goals: removing redundant data (such as keeping the same data in multiple tables) and ensuring that data dependencies make sense (only storing related data in a table). Both of these objectives are worthwhile ones because they help a database use less space and make sure the data is stored logically. A database's columns, or attributes, and tables, or relationships, are organized according to a set of normal form rules during normalization. These normal forms act as a kind of check and balance system to preserve the integrity of relationships between the attributes and relations and are what direct the

normalization process. Through a set of guidelines (called "normal forms"), the normalization process seeks to guarantee that the integrity of the database is maintained regardless of whether any data is changed, added, or destroyed. First Normal Form (INF), Second Normal Form (2NF), and Third Normal Form were the initial names given to the suggested normal forms (3NF). (The Basics of Database Normalization, 2022)

The scenario of 3rd normalization is given below:

Label	UNF
1	Customer
2	Lesson
2	Staff
3	Instrument

1NF

Sameer_Customer
Customer_Name (PK)
Customer_Address
Customer_Contact
Customer_D.O.B

Sameer_Lesson
Lesson_ID (PK)
Lesson_type
Lesson_duraion
Lesson_Fee
Lesson_Paid
Staff_ID
Staff_Name

Sameer_Instrument
Instrument_Name (PK)
Instrument_Taught

2NF

Sameer_Customer
Customer_Name (PK)
Customer_Adderess
Customer_Contact
Customer_D.O.B

Sameer_Lesson
Lesson_ID (PK)
Lesson_Type
Lesson_Duration
Lesson_Fee

Sameer_Instrument
Instrument_Name (PK)
Instrument_Taught

Sameer_Staff
Staff_Id (PK)
Staff_Name

3NF

Sameer_Customer
Customer_Name (PK)
Customer_Adderess
Customer_Contact
Customer_D.O.B

Sameer_Lesson
Lesson_ID (PK)
Lesson_Type
Lesson_Duration
Lesson_Fee

Sameer_Instrument
Instrument_Name (PK)
Instrument_Taught

Sameer_Staff
Staff_Id (PK)
Staff_Name

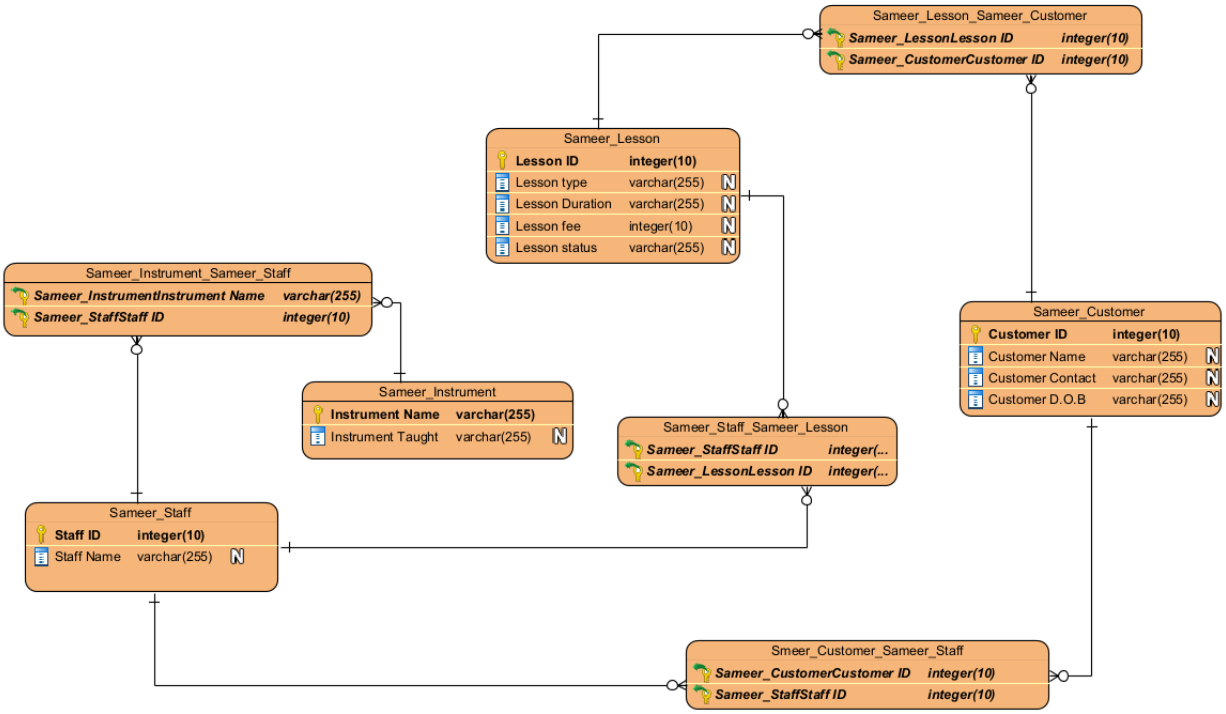
Sameer_Lesson_Customer
Lesson_Lesson_ID (FK)
Customer_Customer_ID (Fk)

Sameer_Staff_Lesson
Lesson_Lesson_ID (FK)
Staff_Staff_ID (FK)

Sameer_Instrument_Staff
Instrumet_Instrument (FK)
Staff_Staff (FK)

Sameer_Customer_Staff
Customer_Customer_ID (FK)
Staff_Staff (FK)

Task 2



An Entity Relationship (ER) Diagram is a kind of flowchart that demonstrates the connections within "entities" like people, things, or concepts within a system. ER Diagrams are most widely used in the fields of software engineering, business information systems, education, and research to construct or debug relational databases. They are also known as ERDs or ER Models, and they use a specified variety of symbols to represent the connectivity of entities, relationships, and their qualities. These symbols include rectangles, diamonds, ovals, and connecting lines. They have verbs for relationships and nouns for entities, mirroring the grammatical framework.

Database design diagrams (DSDs), which highlight relations between elements within entities rather than relationships between things themselves, are linked to ER diagrams. ER

diagrams are usually mixed with data flow diagrams (DFDs), which depict the information flow for systems or processes. (What is an Entity Relationship Diagram (ERD)?, 2022)

Task 3

Meta data of Sameer_Lesson				
Attributes	Data Type	Data length	Key	Description
Lesson ID	Integer	10	Primary key	Primary key of Sameer_Lesson
Lesson type	Varchar	255		
Lesson Duration	Integer	255		
Lesson Fee	Integer	10		
Lesson status	Varchar	255		

Meta data of Sameer_Staff				
Attributes	Data Type	Data length	Key	Description
Staff ID	Intiger	10	Primary key	Primary Key of Sameer_Staff
Staff name	Varchar	255		

Meta data of Sameer_Customer				
Attributes	Data Type	Data length	Key	Description
Customer ID	Intiger	10	Primary Key	Primary key of Customer
Customer Name	Varchar	255		
Customer Contact	Varchar	255		
Customer D.O.B	Varchar	255		

Meta data of Sameer_Lesson_Staff				
Attributes	Data Type	Data length	Key	Description
Sameer_Staff_Staff ID	Intiger	10	Foreign key	Foregin key
Sameer_Lesson_Lesson	Intiger	10	Foreign key	Foregin key

Meta data of Sameer_Instrument				
Attributes	Data Type	Data length	Key	Description
Instrument Name	Varchar	255	Primary key	Primary key of Sameer_Instrument
Instrument Taught	Varchar	255	Primary key	Primary key of Sameer_Instrument

Meta data of Sameer_Instrument_Staff				
Attributes	Data Type	Data length	Key	Description
Sameer_Staff Staff ID	Intiger	10	Foreign key	Foregin key
Sameer_Instrument Instrument Na	Varchar	255	Foreign key	Foregin key

Meta data of Sameer_Lesson_Customer				
Attributes	Data Type	Data length	Key	Description
Sameer_LessonLesson ID	Intiger	10	Foreign key	Foreign key
Sameer_CustomerCustomer	Intiger	10	Foreign key	Foreign key

Meta data of Sameer_Customer_Staff				
Attributes	Data Type	Data length	Key	Description
Sameer_CustomerCustomer ID	Intiger	10	Foreign key	Foreign key
Sameer_StaffStaff	Intiger	10	Foreign key	Foreign key

Task 4

Database creation:

```
create database sameer_cw;
use sameer_cw;
```

Creating customer table:

```
create table sameer_custsomer(
cust_Id int auto_increment,
cust_name varchar(225),
cust_contact varchar(225),
cust_dob varchar(225),
primary key(cust_id)
);
```

Inserting values in customer table:

```
11 •  insert into sameer_customer values
12    (890,"Robert WardPerkins","020 8509 9876","01-Jan-1980"),
13    (891,"Iain Pears","96876885","01-Jan-1981"),
14    (892,"Gene Priest","7567486","01-Jan-1985")
15    ;
16 •  select * from sameer_customer;
17
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Ce

cust_Id	cust_name	cust_contact	cust_dob
890	Robert WardPerkins	020 8509 9876	01-Jan-1980
891	Iain Pears	96876885	01-Jan-1981
892	Gene Priest	7567486	01-Jan-1985
NULL	NULL	NULL	NULL

Creating staff table:

```
create table sameer_staff(
staff_Id varchar(225),
staff_name varchar(225),
primary key(staff_id)
);
```

Inserting values in staff table:

```
24 •  insert into sameer_staff values
25    ("s102","Tanya Silverman"),
26    ("s105","Jason Smith "),
27    ("s204","Cathy Edwards  "),
28    ("S333 ","Roberto Watts  "),
29    ("s405 ","Fatima Khan   "),
30    ("s431","Idris Shah"),
31    ("s567","Ben Watkins ")
32    ;
33 •  select * from sameer_staff;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

staff_Id	staff_name
s102	Tanya Smith
s105	Jason Smith
s204	Cathy Edwards
S333	Roberto Watts
s405	Fatima Khan
s567	Ben Watkins

Creating lesson table:

```
create table sameer_lesson(
lesson_Id int,
lesson_type varchar(225),
lesson_duration varchar(225),
lesson_fee int,
lesson_status varchar(225),
cust_Id int,
foreign key (cust_Id) references sameer_customer(cust_Id),
primary key(lesson_Id)
);
```

Inserting values in lesson table:

```
48 •  insert into sameer_lesson (lesson_Id,lesson_type,lesson_duration,lesson_fee,lesson_status,cust_Id)values
49    (5221,"Advanced Guitar","1 hour", 20, "individual",890),
50    (5422,"Intermediate Saxophone","2 hour", 15, "individual",890),
51    (7899,"Advanced Guitar ","1 hour", 25, "individual",891)
52    ;
53 •  select* from sameer_lesson;
54
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

lesson_Id	lesson_type	lesson_duration	lesson_fee	lesson_status	cust_Id
5221	Advanced Guitar	1 hour	20	individual	890
5422	Intermediate Saxophone	2 hour	15	individual	890
7899	Advanced Guitar	1 hour	25	individual	891
NULL	NULL	NULL	NULL	NULL	NULL

Creating instrument table:

```
create table sameer_instrument(
instrument_taught varchar(225),
staff_Id varchar(225) references sameer_staff(staff_Id)
);
```

Inserting values in instrument table:

```
69 •  insert into sameer_instrument (instrument_taught,staff_Id)values
70    ("drumming","s102"),
71    ("guitar","s105"),
72    ("saxophone","s204"),
73    ("cello","s333"),
74    ("violin","s405"),
75    ("guitar","s431"),
76    ("electric bass","s567"),
77    ("guitar","s567")
78    ;
79 •   select * from sameer_instrument;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

instrument_taught	staff_Id
drumming	s102
guitar	s105
saxophone	s204
cello	s333
violin	s405
guitar	s431
electric bass	s567
guitar	s567

Task5

1. Query to select lessons taken by Robert Ward-Perkins:

```
60 •  select sameer_customer.cust_name,sameer_lesson.lesson_type from sameer_customer
61     left join sameer_lesson on sameer_customer.cust_Id=sameer_lesson.cust_id ;
62
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

cust_name	lesson_type
Robert WardPerkins	Advanced Guitar
Robert WardPerkins	Intermediate Saxophone

2. Query to select all the customer details for individual guitar lessons:

```
60 •  select sameer_customer.cust_name,sameer_lesson.lesson_type from sameer_customer
61     left join sameer_lesson on sameer_customer.cust_Id=sameer_lesson.cust_id ;
62
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

cust_name	lesson_type
Robert WardPerkins	Advanced Guitar
Robert WardPerkins	Intermediate Saxophone

3. Query to count all the guitar teachers:

```
88 •  SELECT COUNT("guitarTeachers")FROM sameer_instrument WHERE instrument_taught = "guitar";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COUNT("guitarTeachers")
3

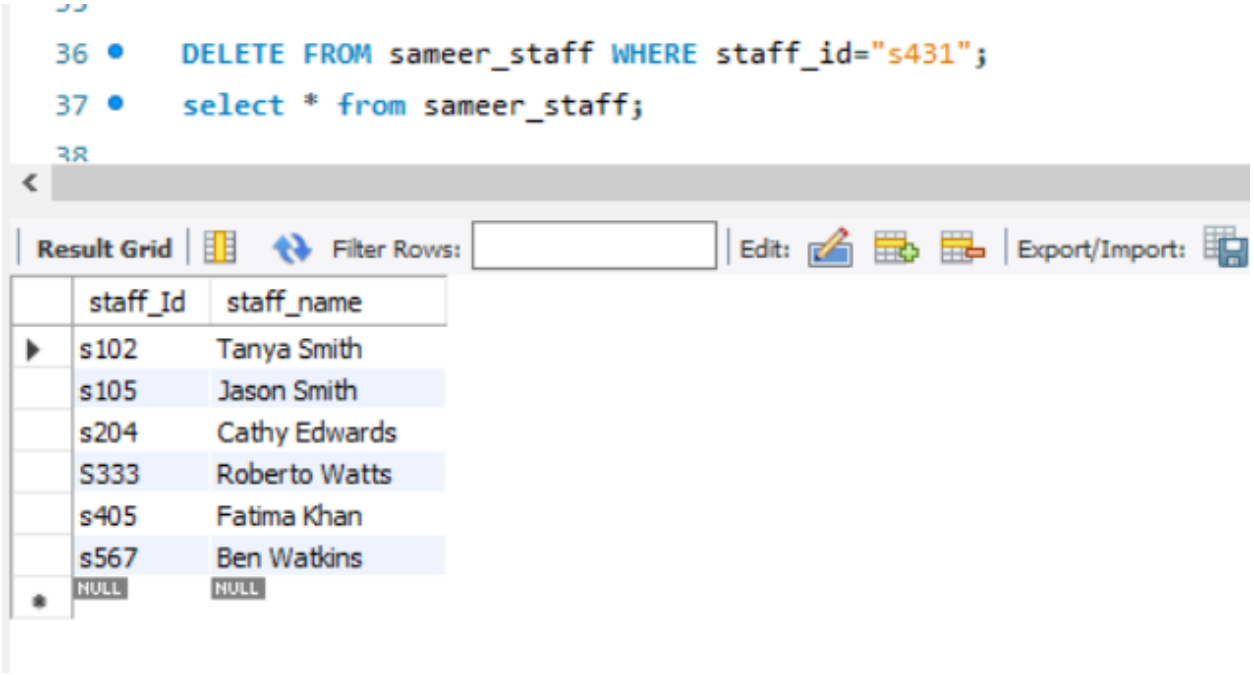
4. Query to update the staff table so that Tanya Silverman is now known as Tanya Smith:

```
39 •  UPDATE sameer_staff SET staff_name = 'Tanya Smith' WHERE staff_Id = "s102";
40
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Conte

staff_Id	staff_name
s102	Tanya Smith
s105	Jason Smith
s204	Cathy Edwards
S333	Roberto Watts
s405	Fatima Khan
s431	Idris Shah
s567	Ben Watkins
NULL	NULL

5. Query to delete staff record for Idris Shah:



Task 6

A method called normalization is used to fix a database's design problems. A database that has been improperly constructed is inconsistent and causes problems when adding, deleting, or changing data. Normalization is required to resolve the database when we insert the data which do not exist. It is important to delete poor design and poor data in database. Data redundancy is not a problem in a normalized table. The data is properly stored in the right table, guaranteeing normalization. A well-designed database guarantees that modifications to one table or field won't have an impact on the others. Through normalization, this is accomplished. If a record is left out during updating, it may result in inconsistent data; normalization fixes this problem and assures data consistency.

I fixed database abnormalities and duplicate data. I have also avoided setting up and updating any unnecessary data dependencies and connections. I stopped data from being deleted

without permission. I made storage space-efficient. When new types of data must be introduced, I decreased the time and complexity of database checks. I have made it easier for people and programs that use databases to obtain and comprehend data. In the process I have used till 3NF for normalization where 1NF eliminates duplications and makes distinct tables for collections of similar data. In 2NF, Elements of data that are present in numerous rows of a table are removed, and new tables with connections between them are created instead. In 3NF, columns that are not reliant on the primary key value are deleted.

Task 7

In its simplest form, data storage refers to the digital recording of files and documents and their subsequent saving in a storage system. Storage systems may use electromagnetic, optical, or other media to keep the data safe and recover it if necessary. File backup and recovery are made simple by data storage in the case of an unanticipated computer failure or cyberattack.

Physical hard drives, disk drives, USB drives, and virtually on the cloud are all options for data storage. It's crucial that your files are protected and accessible in case your systems ever experience a catastrophic failure. Reliability, the robustness of the security measures, and the pricing are some of the most critical aspects to take into account when choosing a data storage solution.

Storage Directly Attached (DAS): It imply, refers to several forms of data storage that are physically attached to your computer. Usually, only one machine has access to this storage. Among the common gadgets in this category are

1. Disk drive
2. The Solid-State Drive (SSD)

3. floppy disks

Advantage of DAS

- simple and widely available to use
- performs far better than NAS and SAN

Disadvantage of DAS

One drawback is that only the apps operating on the specific server or desktop machine can directly access the data; other user groups are not able to do so. Additionally, DAS lacks the necessary network hardware and operating system to enable independent sharing of storage resources.

Network attached storage (NAS): It is a server for file-level data storage that is linked to a computer network. It gives different groups of clients and permitted network users access to and storage of data from a central place. Common shared applications supported by these systems include data logging, email systems, video recording and editing, business analytics, financial records, and many more.

Advantage of NAS

NAS systems are easy to use, which is good for small business owners. NAS is a cost-effective data backup solution that is simple and safe, and it can replace DAS (direct-attached storage). Additionally, compared to other storage solutions like DAS or SAN, it greatly lowers unused space (storage area network). Additionally, because NAS systems are always available, it is simple for staff members to work together, serve clients, and enable collaborative development initiatives. The NAS system functions similarly to the Cloud in that it may be accessed from a distance with a network connection. As a result, the personnel can work whenever they choose from any location.

Disadvantage of NAS

Scale and performance are two areas where NAS falls short. The NAS appliance's resources are limited, and as more users request access, it will become unable to keep up, resulting in sluggish performance and user annoyance. Due to the difficulties of low throughput and high latency, NAS systems cannot be easily scaled up or out, and NAS protocols like Network File System (NFS) and SMB are too slow for high-performance applications. Additionally, NAS depends on networks since files are shared through local area networks (LAN). Data packets are used by the LAN to convey data from one location to another by being divided into many segments and sent to any terminals. (What is Data Storage, 2022)

Reference

Lifewire. 2022. *The Basics of Database Normalization*. [online] Available at: https://www.lifewire.com/database-normalization-basics-1019735 [Accessed 27 July 2022].

Lucidchart. 2022. *What is an Entity Relationship Diagram (ERD)?*. [online] Available at: https://www.lucidchart.com/pages/er-diagrams [Accessed 27 July 2022].

Cdw.com. 2022. *What is Data Storage*. [online] Available at: https://www.cdw.com/content/cdw/en/articles/datacenter/what-is-data-storage.html [Accessed 27 July 2022].