

**DipIT07 – Introduction to Database System**

**TOPIC: Course work**

**Student Name:** Naramee Chakhun

**Student Id: NP03A180198**

**Submitted date:** 2019/06/02

**Submitted to:** Deepson Shrestha

Acknowledgement

I want to give my thanks to our Module Leader Mr. Prakash Shrestha and Lecturer Mr. Deepson Shrestha who gave me the golden opportunity to do this report which helped me in doing a lot of Research and improving my research skills. I came to know about so many new things I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time.

**Marking Guidelines**

This coursework accounts for 50% of your total module.

The marking Scheme for the java program shall be as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN. | Title | Marks | Comments |
| A 1.1 | Appropriateness of Research | /5 |  |
| A 1.2 | Proper Listing of Clients of Researched Database | /10 |  |
| A 2.1 | 1NF Description and Example | /10 |  |
| A 2.2 | 2NF Description and Example | /5 |  |
| A 2.3 | 3NF Description and Example |  |  |
| B 1.1 | Identification of Entities and Attributes of given scenario | /15 |  |
| B 1.2 | PK and FK Creation and Referential Integrity | /5 |  |
| C 1.1 | ER Diagram | /10 |  |
| D 1.1 | Proper use of Insert and Create Command | /10 |  |
| D 1.2 | Proper use of Select command | /5 |  |
| D 1.3 | Proper use of Select command | /5 |  |
| D 1.4 | Proper use of Select command | /5 |  |
| D 1.5 | Proper use of Select command | /5 |  |
| D 1.6 | Proper use of Select and Group function command | /10 |  |
| E | Viva | A/B/F | \*F means fail in the overall coursework |

Contents

[A. Research 6](#_Toc10210120)

[B. Relational Model 11](#_Toc10210121)

[C. DBMS 11](#_Toc10210122)

[D. SQL 11](#_Toc10210123)

[E. References 11](#_Toc10210124)

# Research

1. Describe briefly, how MySQL is suitable for the small and medium size of organization? Also identify the five major clients of it.

MySQL is a proven and cost-effective database solution that will help reduce the cost of your database software infrastructure by over 90%. It cuts systems downtime by 60%. It lowers the hardware expenditure by 70%. It reduces the administration, engineering and support costs by up to 50%. It doesn't just highlight the problems also supply methods of how to fix it. It has made a significant improvement in the performance of some of our slower queries.

The five major clients of MySQL are:

* Alcatel.Lucent
* Github
* Booking.com
* Ericsson
* Lufthansa

(Oracle Corporation, n.d.)

2. By the help of real example describe the 1NF, 2NF & 3NF in detail.

* **Normalization** is a process of organizing the data in database to avoid data redundancy, insertion anomaly, updates anomaly & deletion anomaly.

The most used normal forms are:

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)
4. First Normal Form (1NF): As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

**Example**: Suppose a company wants to store the names and contact details of its employees. It creates a table that looks like this:

|  |  |  |  |
| --- | --- | --- | --- |
| Emp\_id | Emp\_Name | Emp\_Address | Emp\_Phone |
| 01 | Nancy | New York | 7654390034 |
| 02 | Harry | London | 8765905432 |
| 02 | Harry | London | 8743290453 |
| 03 | Peter | Argentina | 5432178908 |
| 04 | Amelia | Paris | 6758943210 |
| 04 | Amelia | Paris | 6759987043 |

1. Second Normal Form (2NF): A table is said to be in 2NF if both the following conditions hold:

* Table is in 1NF (First normal form)
* No non-prime attribute is dependent on the proper subset of any candidate key of table.

**Example**: Suppose a school wants to store the data of teachers and the subjects they teach. They create a table that looks like this: Since a teacher can teach more than one subjects, the table can have multiple rows for a same teacher.

|  |  |  |
| --- | --- | --- |
| Teacher\_ID | Subject | Teacher\_Age |
| 1 | Mathematics | 45 |
| 1 | Physics | 45 |
| 2 | Biology | 45 |
| 3 | Physics | 36 |
| 3 | Chemistry | 36 |

**Candidate Keys: {Teacher\_ID, Subject}**  
**Non prime attribute**: Teacher\_Age

The table is in 1 NF because each attribute has atomic values. However, it is not in 2NF because non prime attribute Teacher\_Age is dependent on Teacher\_ID alone which is a proper subset of candidate key. This violates the rule for 2NF as the rule says “**no** non-prime attribute is dependent on the proper subset of any candidate key of the table”.

To make the table complies with 2NF we can break it in two tables like this:  
**Teacher\_Details table:**

|  |  |
| --- | --- |
| Teacher\_ID | Teacher\_Age |
| 1 | 45 |
| 2 | 45 |
| 3 | 36 |

**Teacher\_Subject table:**

|  |  |
| --- | --- |
| Teacher\_ID | Subject |
| 1 | Mathematics |
| 1 | Physics |
| 2 | Biology |
| 3 | Physics |
| 3 | Chemistry |

Now the tables comply with Second normal form (2NF).

Third Normal Form (3NF): A table design is said to be in 3NF if both the following conditions hold:

Table must be in 2NF

[Transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) of non-prime attribute on any super key should be removed.

An attribute that is not part of any [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:

X is a [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) of table

Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as prime attribute.

Example: Suppose a company wants to store the complete address of each employee, they create a table named Employee\_Details that looks like this:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Emp\_id | Emp\_Name | Emp\_Zip | Emp\_State | Emp\_City |
| A1 | Menasha | 233784 | New York | New York |
| A2 | George | 244577 | California | Los Angeles |
| A3 | Zen | 214599 | Texas | Houston |
| A4 | Sian | 245908 | Ohio | Columbus |
| A5 | Isa | 267490 | Arizona | Phoenix |

**Super keys**: {Emp\_id}, {Emp\_id, Emp\_Name}, {Emp\_id, Emp\_Name, Emp\_Zip}  
**Candidate Keys**: {Emp\_id}  
**Non-prime attributes**: all attributes except Emp\_id are non-prime as they are not part of any candidate keys.

Here, Emp\_State & Emp\_City dependent on Emp\_Zip. And, Emp\_Zip is dependent on Emp\_id that makes non-prime attributes (Emp\_State & Emp\_City) transitively dependent on super key (Emp\_id). This violates the rule of 3NF.

To make this table complies with 3NF we have to break the table into two tables to remove the transitive dependency:

**Employee table:**

|  |  |  |
| --- | --- | --- |
| Emp\_id | Emp\_Name | Emp\_Zip |
| A1 | Menasha | 233784 |
| A2 | George | 244577 |
| A3 | Zen | 214599 |
| A4 | Sian | 245908 |
| A5 | Isa | 267490 |

**Emp\_Zip table:**

|  |  |  |
| --- | --- | --- |
| Emp\_Zip | Emp\_State | Emp\_City |
| 233784 | New York | New York |
| 244577 | California | Los Angeles |
| 214599 | Texas | Houston |
| 245908 | Ohio | Columbus |
| 267490 | Arizona | Phoenix |

# (Singh, n.d.)

# Relational Model

1. There are forty students in a class studying BSc Computer Science. Some of them are working on the different projects as extra employment. Identify the possible entity and attributes of this scenario (at least 10 attributes for each).
2. Identify Entities and Attributes for above scenario.
3. Identify PK & FK constraints.

# DBMS

1. Draw the ER diagram of above described scenario.

# SQL

1. Create the tables and insert at least 10 records for each.
2. Display the names of students in Alphabet Order.
3. Count the total number of employments.
4. Display the Project Name and Address of the student whose name is starting with ‘K’.
5. Select all information about students who belongs to Pokhara, Kathmandu and Gorkha.
6. Identify the total salaries paid to each student from each project.

# References

Oracle Corporation, n.d. *Oracle corporation.* [Online]   
Available at: http//www.mysql.com/industry/faq  
[Accessed 23 05 2019].

Singh, C., n.d. *Beginners Book.* [Online]   
Available at: http//www.beginnersbook.com  
[Accessed 23 05 2019].