Assignment-2

COIT20247

Database Design and Development

Term -1, 2020

Lecturer/Tutor:

Annie Ding

Prepared By:

Bishal Budhathoki

12116421

**Table of Contents**

[1.A) ERD Mapping into set of relations: 2](#_Toc41337975)

[1.B) Functional dependency 2](#_Toc41337976)

[Data Integrity: 3](#_Toc41337977)

[Report of Implementation: 4](#_Toc41337978)

# 1.A) ERD Mapping into set of relations:

CUSTOMER (*CustomerID*, CustomerName, CustomerContactNum, StreetAddress, City, PostCode)

EMPLOYEE (*EmployeeId*, EmployeeName, EmployeeStartDate)

JOB\_REQUEST *(JobId*, JobRequestDate, DateScheduled, JobStartDate, JobEndDate, JobType, TotalCharges, *CustomerId*)

Foreign key (*CustomerId*) references CUSTOMER

INSTALLATION (*InstallationId*, *JobId*, *AcUnitId*, HP, Address, InstallationType)

Foreign key (*JobId*) references JOB\_REQUEST

Foreign key (*AcUnitId*) references AC\_UNIT

REPAIR (*JobId*, Problem, *InstallationID*)

Foreign key (*InstallationID*) references INSTALLATION

Foreign key (*JobId*) references JOB\_REQUEST

WALL\_WINDOW\_MOUNTED (*InstallationId*, location)

Foreign key (*InstallationId*) references INSTALLATION

DUCTED (*InstallationId*, *ZoneId*)

Foreign key (*InstallationId*) references INSTALLATION

Foreign key (*ZoneId*) references ZONE

ZONE (*ZoneId*, ZoneName, NumOutlets)

AC\_UNIT *(AcUnitId*, Make, Model, HP)

PAYMENT (*PaymentId*, PaymentDate, PaymentAmount, *InvoiceId*)

Foreign key (*InvoiceID*) references INVOICE

INVOICE (*InvoiceId*, InvoiceDate, InvoiceAmount, *JobId*)

Foreign key (*JobId*) references INVOICE

EMP\_JOB *(EmployeeId,* EmpStartDate, EmpFinishDate, *JobId*)

Foreign key (*EmployeeId*) references EMPLOYEE

Foreign key (*JobId*) references JOB\_REQUEST

# 1.B) Functional dependency

1. CustomerId🡪 CustomerName, CustomerContactNum, StreetAddress, City, PostCode

CUSTOMER (*CustomerId*, CustomerName, CustomerContactNum, StreetAddress, City, PostCode)

* + 1. Here, Primary key is CustomerId which is specific for every customer. Here only one customer name for every customer. Customer can have only one address here even if they have multiple address. There can be one phone-number for every individual. There are no repeating groups, this relation follows 1NF rule.
    2. Primary key is CustomerId. Candidate key is CustomerId as name is not always unique. Address may not be unique all the time. Contact number is always unique. CustomerId – candidate key is simple with single valued attribute with no partial dependencies are possible. So, it completes 2NF rules.
    3. In the table, name of customer cannot determine attribute functionally as two customers can share same name. Similarly, goes for address and contact number. So, there is no transitive dependencies in the table. As, transitive dependencies is not found, it completes 3NF rules.

1. EmployeeId🡪EmployeeName, StartDate

EMPLOYEE (*EmployeeId*, EmployeeName, EmployeeStartDate)

1. EmployeeId - primary key validates the uniqueness of employee, meaning there can be one employee per EmployeeId because as per 1 NF rule only one record to be stored at a time. So, since we ensured there are no repeating groups it is in 1NF. There will be only one start date for each employee.

ii. Here EmployeeId is primary key and it is the candidate key. Employees name cannot always be unique and their start date can be same as more than one employee can commence at a same date. So, EmployeeId - candidate key is simple and single valued attribute with no partial dependencies. So, it completes 2NF rules.

iii. In the table, name of employee cannot determine attribute functionally as two or more employee can share same name and start date can also be same. As, transitive dependencies is not found, it completes 3NF rules.

# Data Integrity:

For this Employee table is selected.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Integrity Constraint Implemented** | **Error Message** |
| EmployeeId | AutoNumber | Indexed = Yes | The EmployeeId do not accept same values multiple time. |
| EmployeeName | Short Text | Required = Yes | Null value is not accepted |
| EmployeeStartDate | Date/Time | [EmployeeStartDate]<=Date() | Employee date to join / start should not be upcoming / future date. |

# Report of Implementation:

After successfully completing this assignment, I obtained knowledge on using Microsoft Access 2016 and how to create table using the option able in the access as well as using query and add primary key to it as well as foreign key and ER Diagram. Along with that, I also learned to prepare report and show relationships between tables in Access. I got to learn a bit about MySQL Workbench too.

Access was an application that I had never used so I did face some issue while using it like while navigating through different options and functionality but as I kept on using it and direction from tutor, I got used to it and navigation became easy. Creating relations between tables was tough for me to understand as after creating table and relations between them, when I change the validation rule then relations are broken else validation rule would not apply on attributes of table entity. Then I selected table again and do the relation between table manually. At first, I felt like was doing wrong and took a while to figure out that it happens due to change in validation rule. Tutor – Annie Ding was available for queries too. Slides became very useful when doing relation and queries of the given assignment.

I have completed this assignment using available materials from Moodle of COIT20247 only.