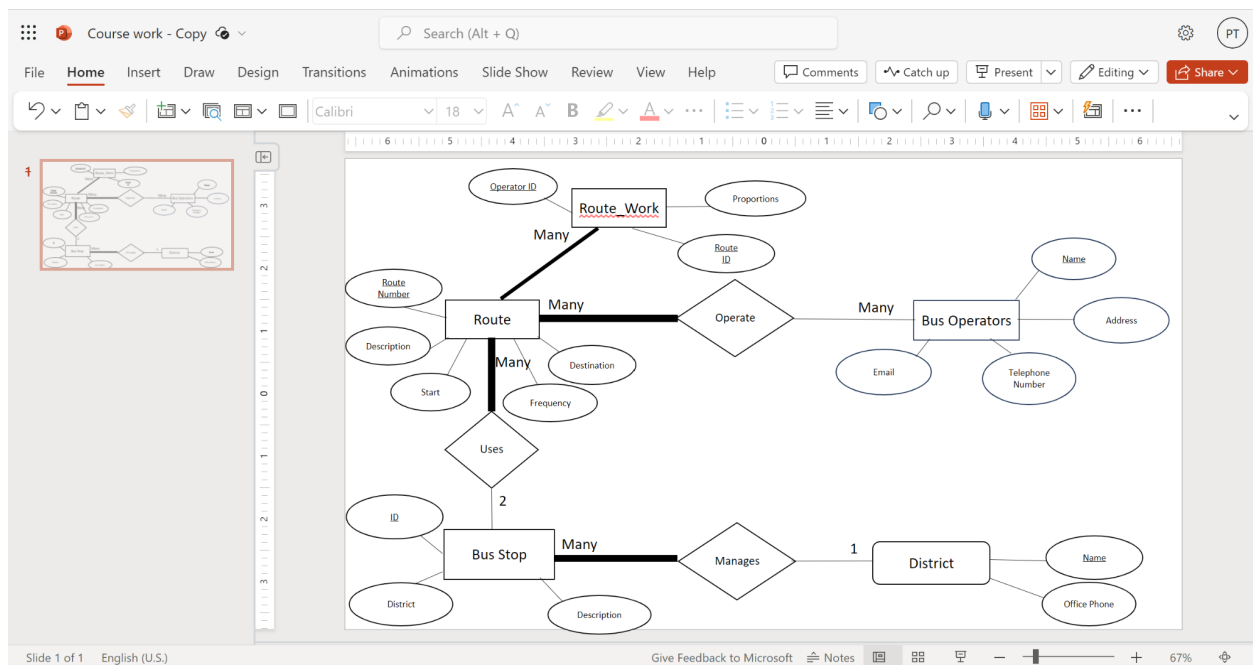


CSC 1033 Semester-1 Coursework

Student name- Phone Myat Thu

Student No. - 230668418

Task 1



Task 2

Bus Operator Table

```
CREATE TABLE `BusOperators` (  
  `Name` varchar(255) NOT NULL,  
  `Address` varchar(255) DEFAULT NULL,  
  `Email` varchar(100) DEFAULT NULL,  
  `TelephoneNumber` varchar(20) DEFAULT NULL,  
  PRIMARY KEY (`Name`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Expalnation: For the bus operator table, I had created it for the new bus operator entity. I put attributes such as name, address, telephone number and email with the suitable maximum value. It operates the route and its primary key is the name which is unique.

Route Table

```

CREATE TABLE `Route` (
  `RouteNumber` varchar(255) NOT NULL,
  `Frequency` int(11) DEFAULT NULL,
  `Start` varchar(255) DEFAULT NULL,
  `Destination` varchar(255) DEFAULT NULL,
  `Description` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`RouteNumber`),
  KEY `Route_FK` (`Start`),
  KEY `Route_FK_1` (`Destination`),
  CONSTRAINT `Route_FK` FOREIGN KEY (`Start`) REFERENCES `BusStop` (`ID`),
  CONSTRAINT `Route_FK_1` FOREIGN KEY (`Destination`) REFERENCES `BusStop` (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Explanation: For the route table which is operated by the bus operators and its primary key is route number and foreign key are Stop and Destination which are connected with ID' from the bus stop table and new attributes such as route number, frequency that shows the number of buses per hour and description how the buses go clockwise or anticlockwise including two remaining attributes that designate the start and destination of the journey.

Bus Stop Table

```

CREATE TABLE `BusStop` (
  `ID` varchar(255) NOT NULL,
  `District` varchar(100) DEFAULT NULL,
  `Description` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `BusStop_FK` (`District`),
  CONSTRAINT `BusStop_FK` FOREIGN KEY (`District`) REFERENCES `District` (`Name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Explanation: For the bus stop table which is used by routes has ID as the primary key and District as the foreign key which is connected by 'Name' from the District table and there are no new attributes and only used the remaining attributes such as ID, District and Description.

District Table

```

CREATE TABLE `District` (
  `Name` varchar(100) NOT NULL,
  `OfficePhone` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`Name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Explanation: For the district table which manages bus stops has no new attributes and only used Name and Office Phone that were given. This table has Name as a primary key.

Route_Work Table

```

CREATE TABLE `Route_Work` (

```

```

`RouteID` varchar(255) NOT NULL,
`OperatorID` varchar(255) NOT NULL,
`Proportions` varchar(100) DEFAULT NULL,
PRIMARY KEY (`RouteID`,`OperatorID`),
KEY `Route_Work_FK` (`OperatorID`),
CONSTRAINT `Route_Work_FK` FOREIGN KEY (`OperatorID`) REFERENCES `BusOperators`
(`Name`),
CONSTRAINT `Route_Work_FK_1` FOREIGN KEY (`RouteID`) REFERENCES `Route`
(`RouteNumber`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Explanation: For the Route_Work table that connects with the Route table, the Route_Work table has the attributes such as RouteID, OperatorID and Proportions which shows the numbers of buses per hour. The primary keys are the RouteID and OperatorID and also foreign keys as the OperatorID which is connected with Name from the BusOperators table and RouteID which is connected with Route from RouteNumber table.

Task 3

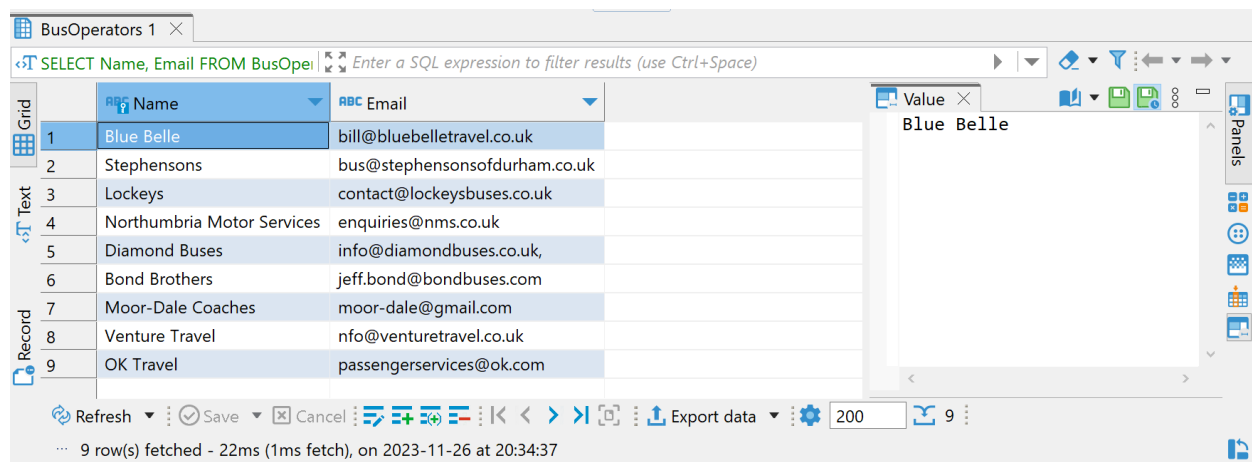
(a)

```

SELECT Name, Email
FROM BusOperators
ORDER BY Email ASC;

```

Ans: OK Travel has the email address which comes last if all the email addresses were ordered alphabetically.



	Name	Email
1	Blue Belle	bill@bluebeltravel.co.uk
2	Stephensons	bus@stephensonsofduham.co.uk
3	Lockeys	contact@lockeysbuses.co.uk
4	Northumbria Motor Services	enquiries@nms.co.uk
5	Diamond Buses	info@diamondbuses.co.uk,
6	Bond Brothers	jeff.bond@bondbuses.com
7	Moor-Dale Coaches	moor-dale@gmail.com
8	Venture Travel	nfo@venturetravel.co.uk
9	OK Travel	passengerservices@ok.com

(b)

```

SELECT District, Count(*) As NumStops
FROM BusStop
GROUP BY District;

```

Ans: Central has 3 stops, Northern has 5 stops and Southern has 4 stops.

BusStop 1

SELECT District, Count(*) As NumStops

	District	NumStops
1	Central	3
2	Northern	5
3	Southern	4

Value

Edit value:

Central

Dictionary (District): (Define Description)

Type part of dictionary value to search

Value	Description
Central	0191 266 5494
Northern	01670 111 222
Southern	0191 377 6611

Refresh Save Cancel Export data 200 3

3 row(s) fetched - 19ms, on 2023-11-26 at 14:48:13

(c)

```
SELECT DISTINCT d.Name, d.OfficePhone
FROM Route r
JOIN BusStop bs ON r.`Start` = bs.ID OR r.Destination = bs.ID
JOIN District d ON bs.District = d.Name
WHERE r.RouteNumber = '16a'
```

Ans: if someone from 16a notices that the light was broken at the bus shelter when they boarded and the litter bin had been vandalized at the terminus, they can call **0191 377 6611** from **Southern** district.

District 1

SELECT DISTINCT d.Name, d.OfficeP

	Name	OfficePhone
1	Southern	0191 377 6611

Value

Southern

Refresh Save Cancel Export data 200 1

1 row(s) fetched - 20ms, on 2023-11-26 at 20:32:47

(d)

```
SELECT DISTINCT bo.Email
FROM BusOperators bo
JOIN Route_Work rw ON bo.Name = rw.OperatorID
JOIN Route r ON rw.RouteID = r.RouteNumber
```

```
JOIN BusStop bs ON r.Start = bs.ID OR r.Destination = bs.ID
WHERE bs.Description LIKE '%Estate%'
ORDER BY bo.Email ASC;
```

Ans: The list of email address that bus stop sign contained the word “Estate” in its name are

- 1) bill@bluebelletravel.co.uk
- 2) bus@stephensonsofdurham.co.uk
- 3) contact@lockeysbuses.co.uk
- 4) enquiries@nms.co.uk
- 5) moor-dale@gmail.com
- 6) passengerservices@ok.com

The screenshot shows a database application window titled 'BusOperators 1'. The SQL query entered is 'SELECT DISTINCT bo.Email FROM Bt'. The results are displayed in a table with 6 rows. A 'Value' popup is visible over the first row, showing the email address 'bill@bluebelletravel.co.uk'. The bottom status bar indicates '6 row(s) fetched - 96ms, on 2023-11-27 at 13:18:27'.

Grid	RBC Email
1	bill@bluebelletravel.co.uk
2	bus@stephensonsofdurham.co.uk
3	contact@lockeysbuses.co.uk
4	enquiries@nms.co.uk
5	moor-dale@gmail.com
6	passengerservices@ok.com

Task 4

(a)

A recurring group is the new attribute in District that lists all of the staff members who work there as well as some critical details about them which means that numerous values for this attribute can exist in a single District record which violates the 1NF rule.

(b)

In this scenario, The first and second normal forms the 1NF and 2NF are broken by the proposed Staff entity because the Staff entity's main key violates 1NF as it uses composite primary keys (Postcode from Office and District from District) and the second 2NF is broken by the composite primary key, which additionally adds dependencies on partial keys from other entities.

(c)

In this scenario, 1NF and 2NF rules are broken because a single member can have different office addresses and postcodes over the time and also the non-key attribute District influences the office address and postcode characteristics. This is because a staff member's office and postcode are decided by the district where they work.

Task 5

(a)

Three examples of the relevant metadata for monitoring the usage of databases are Index Statistics, Query Execution Plans and Database Workload Trends.

Explanation

1) Index Statistics

Index improves the efficiency of assessing the query that is helped by tracking information associated with index statistics such as quantity and frequency usage of index which enhance the speed of the query to be identified by the usage of patterns of or index fragmentation.

2) Query Execution Plans

It is really important to understand the query execution plans. Insights into query performance are provided via metadata that provides information on query execution plans, including execution time, resources spent, and joins made. With the analysis of the execution plans, it can improve the performance of modifying the SQL query by adding additional indexes or organizing the tables.

3) Database Workload Trends

Understanding the user behavior made it easier for metadata to analyze data relating to the database workload trends. This data helps with resource assigning, the scalability of resources during peak periods to protect from deterioration of database access.

(b)

As a database administrator, I can assist the employer in attaining ISO9001 certification by documenting database processes, creating comprehensive documentation for all database related processes which includes data entry, backup, recovery and maintenance procedures. This documentation helps to ensure consistency

in database operations that is essential for QMS. It provides a clear protocol for the employees to follow, reducing the risk of errors and ensuring that all activities are assigned with quality standards. Moreover, it's a way to demonstrate to ISO auditors that your database practices are well-defined and controlled with quality management systems. As an example, by showing how the management of the database aligns with the council's commitment to quality and continuous improvement. By this contribution to overall QMS and supporting the ISO9001 certification processes. It will be the key aspects of attaining the certification for the employer.