MYSQL Assignment

Due Date of submission: 24/10/2023

Andre Watts

2023

Contents

[Task 1 – Types of Relationships 2](#_Toc148519494)

[Task 2 – Normalization 3](#_Toc148519495)

[SQL Tasks 4](#_Toc148519496)

[Task 3: Using count, get the number of cities in the USA 6](#_Toc148519497)

[Task 4: Find out the population and life expectancy for people in Argentina 6](#_Toc148519498)

[Task 5: Using Order By, Limit, what country has the highest life expectancy 7](#_Toc148519499)

[Task 6: Select 25 cities around the world that start with the letter 'F' in a single SQL query 8](#_Toc148519500)

[Task 7: Create SQL statement to display columns ID, Name, Population, From the city table and limit results to first 10 row only 9](#_Toc148519501)

[Task 8: Create a SQL statement to find only those cities from city table whose population is larger than 2,000,000 10](#_Toc148519502)

[Task 9: Create SQL statement to find all city names from city table whose names begins with 'Be' prefix 12](#_Toc148519503)

[Task 10: Create a SQL statement to find only those cities from city table whose population is between 500,000 and 1,000,000 13](#_Toc148519504)

[Task 11: Create SQL statement to find a city with the lowest population in the city table 20](#_Toc148519505)

[Task 13: Creating an EER Diagram 22](#_Toc148519506)

[Task 14: EER Questions 22](#_Toc148519507)

[Reflection 22](#_Toc148519508)

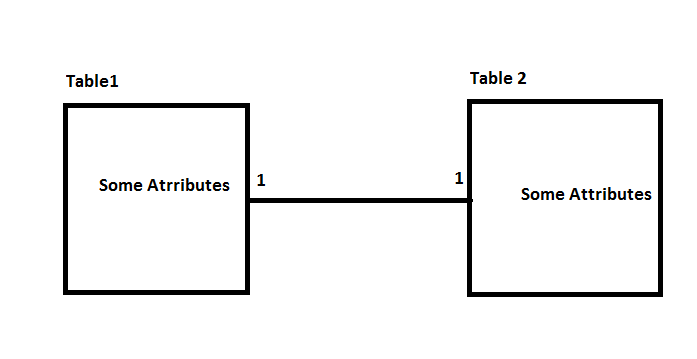
# Task 1 – Types of Relationships

What is a relationship in a relational database? A relationship between two tables in relational database normally contains a foreign key in one of the tables which references the primary key within another table therefore creating a relationship between the two or more tables. An entity-relationship diagram (ERD) can be used as a graphical representation of the relationship between tables in a relational database.

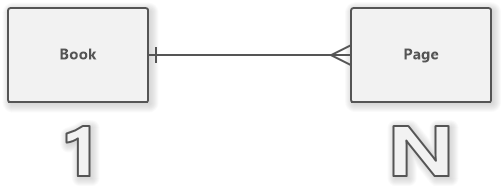
There are 3 main types of relationships in a database.

**One-to-One**

A one-to-one relationship is where a row in table 1 only relates to one row in table 2. An example is a husband and wife. A husband will only have one wife and the wife will only have one husband.

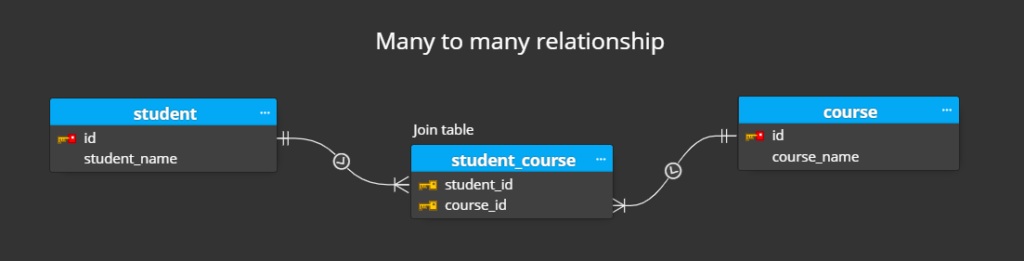


**One-to-Many**

A one-to-many relationship when one row in table 1 relates to more than one row in table 2. However, the rows in table 2 cannot be related to more than one row in table 1. An example is the relationship between a book and the pages within the book. A book has multiple pages but the pages only relate to one book.

**Many-to-Many**

A many-to-many relationship when multiple rows within table 1 relate to multiple rows within table 2. An example is the relationship between products and suppliers. A supplier may deliver many products to a company and the company can order products from many suppliers.



**Problems with One-to-One and Many-to-Many Relationships**

One-to-One relationships are not common in relational databases as table 1 can have only one matching record in table 2 and table 2 can have only one matching record in table 1. Information that is related this way should be presented in one table.

The problems with many-to-many relationships in a relational database is the likelihood of duplication and redundancy of data. This can result in incorrect results and use excessive resources. It can also cause difficulty updating the data and accidental deletion of data.

# Task 2 – Normalization

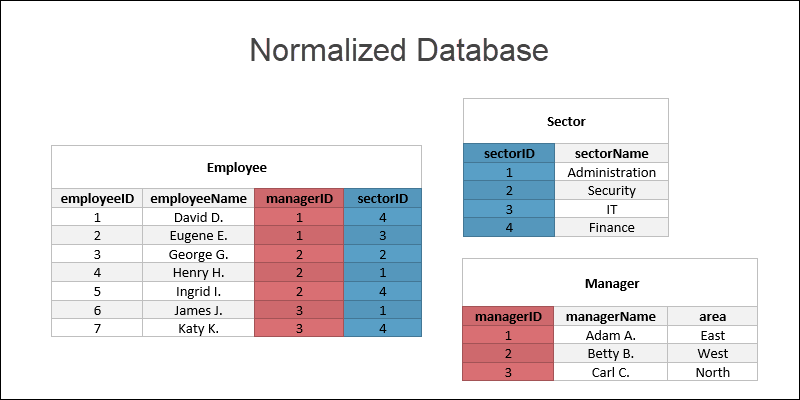
Normalization in relational databases involves organizing the data in a way involves breaking down a table into more than one table and then creating relationships between these tables It is a way to organise the tables in an organised and consistent way.

It is very important to database development because it allows you to avoid data redundancy and maintain integrity of the database. It also helps remove undesirable characteristics when updating and deleting data.

A primary key is used to link tables together. A primary key is a column that uniquely identifies the rows in the data. Most forms of ID such as Customer ID or Product ID can uniquely identify the rows within the database.

A foreign key is a primary key from another table which relates the tables together.

A composite key is like a primary key but instead of one column, it can have multiple columns.



# SQL Tasks

A screenshot of a computer

Description automatically generated

Importing SQL Script and executing script. Refreshing schema to identify the world database.

A screenshot of a computer

Description automatically generated

Action Output after executing the SQL script.

A screenshot of a computer

Description automatically generated

Opening a new query tab using taskbar.

# Task 3: Using count, get the number of cities in the USA

A screenshot of a computer

Description automatically generated

SELECT COUNT(DISTINCT city.Name) AS total\_cities\_USA

FROM city

JOIN country ON city.CountryCode = country.Code

WHERE country.Code = 'USA';

Total number of cities in the USA equals 264.

# Task 4: Find out the population and life expectancy for people in Argentina

A screenshot of a computer

Description automatically generated

SELECT Name, Population, LifeExpectancy

FROM country

WHERE country.Name = 'Argentina';

Population of Argentina = 37032000

Life Expectancy of Argentina = 75.1

# Task 5: Using Order By, Limit, what country has the highest life expectancy

A screenshot of a computer

Description automatically generated

SELECT Name, LifeExpectancy

FROM country

ORDER BY LifeExpectancy DESC

LIMIT 1;

The country with the highest Life Expectancy is Andorra with a Life Expectancy of 83.5.

# Task 6: Select 25 cities around the world that start with the letter 'F' in a single SQL query

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

SELECT \*

FROM city

WHERE Name LIKE 'F%'

ORDER BY Name

LIMIT 25;

25 cities that begin with the letter F ordered by Name.

# Task 7: Create SQL statement to display columns ID, Name, Population, From the city table and limit results to first 10 row only

A screenshot of a computer

Description automatically generated

SELECT ID, Name, Population

FROM city

LIMIT 10;

First 10 rows of the city table showing the ID, Name and Population columns.

# Task 8: Create a SQL statement to find only those cities from city table whose population is larger than 2,000,000

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

SELECT \*

FROM city

WHERE Population > 2000000

ORDER BY Population DESC;

Cities where the Population is > 2,000,000 ordered from highest population to lowest.

# Task 9: Create SQL statement to find all city names from city table whose names begins with 'Be' prefix

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

SELECT \*

FROM city

WHERE Name LIKE 'Be%'

ORDER BY Name;

All cities with ‘Be’ prefix ordered by Name in ascending order.

# Task 10: Create a SQL statement to find only those cities from city table whose population is between 500,000 and 1,000,000

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

SELECT \*

FROM city

WHERE Population BETWEEN 500000 AND 1000000

ORDER BY Population DESC;

All cities where the Population is between 500,000 and 1,000,000 ordered by the highest population to the lowest population.

# Task 11: Create SQL statement to find a city with the lowest population in the city table

A screenshot of a computer

Description automatically generated

SELECT \*

FROM city

ORDER BY Population ASC

LIMIT 1;

The city Adamstown has the lowest population in the city table which is a population of 42.

Task 12: Create a SQL statement to show the population of Switzerland and all the languages spoken there.A screenshot of a computer

Description automatically generated

SELECT country.Name, Population, countrylanguage.Language

FROM country

JOIN countrylanguage

ON country.code = countrylanguage.CountryCode

WHERE country.Name = 'Switzerland';

French, German, Italian and Romansh are all the languages spoken in Switzerland with a population of 7160400.

# Task 13: Creating an EER Diagram

A screenshot of a computer

Description automatically generated

# Task 14: EER Questions

**Identify the primary key in country table –** The primary key is Code CHAR(3)

**Identify the primary key in city table –** The primary key is ID INT

**Identify the primary key in countrylanguage table –** The primary key is Language CHAR(30)

**Identify the foreign key in city table –** The foreign key is CountryCode CHAR(3)

**Identify the foreign key in countrylanguage –** The foreign key is CountryCode CHAR(3)

# Reflection

First, I discussed the different types of relationships within relational databases and provided examples. I also discussed what is Normalization and why it’s important to databases. Then, I used SQL to run queries and complete various tasks which involve finding out specific information about the data.

I didn’t really have any major problems with SQL, but it was my first time using it, so it took a while for me to get used to the various systems and how code is written but eventually, I got used to it and I was able to complete the tasks with no major problems.

The considerations I had was to make sure the code I was writing was clear and formatted in am way that was easy to read. This involves spacing out the tasks and properly labelling the tasks.

I believe my outcomes were good and the results I got made sense within the data. There were no results which were outliers or seemed incorrect at the time.

Finally, I believe I made significant developments on SQL, and I believe I have good knowledge on how to execute various queries and how the code is written in SQL to get the results you desire. Before this I had no knowledge or experience of SQL.