



Data Technician

Name: Nathan Mullings

Course Date: 2025/03/03

Table of contents

Day 1: Task 1	3
Day 1: Task 2	5
Day 3: Task 1	5
Day 4: Task 1: SQL Practical	8
Day 4: Task 2: Written (Optional)	27

Day 1: Task 1

Please research and complete the below questions relating to key concepts of databases.

What is a primary key?	Is a unique identifier in a table eg. Customer id
How does this differ from a secondary key?	A secondary key (also known as a foreign key) links to a primary key from another table, but its self is not a primary key.
How are primary and foreign keys related?	A foreign key links and references a primary key from another table.
Provide a real-world example of a one-to-one relationship	Passport ID & person's full name. each person can have only one passport, and each passport belongs to only one person.
Provide a real-world example of a one-to-many relationship	customer name & order id A customer can place multiple orders. Each order belongs to only one customer.
Provide a real-world example of a many-to-many relationship	Students & Courses A student can enrol in multiple courses. A course can have multiple students.





Day 1: Task 2

Please research and complete the below questions relating to key concepts of databases.

What is the difference between a relational and non-relational database?	<p>Relational database is structured data, with predefined schemas and relationships between tables.</p> <p>Non-relational databases is un-structured data that comes in various formats (e.g., key-value pairs, documents, graphs) that don't require a fixed schema.</p>
What type of data would benefit off the non-relational model?	<p>Social media data would benefit from the non-relational model because it often involves unstructured or semi-structured data (e.g., posts, comments, images, and user interactions) that can vary in format and structure. Non-relational databases allow flexibility in storing and scaling this diverse, evolving data without requiring a predefined schema, making them more suitable for handling the dynamic nature of social media content.</p>
Why?	<p>In short, social media data often includes complex and varied information that fits well with the flexibility of non-relational databases.</p>

Day 3: Task 1

Please research the below 'JOIN' types, explain what they are and provide an example of the types of data it would be used on.

Self-join	<p>A self-join is a type of join in SQL where a table is joined with itself. This is useful when comparing rows within the same table. A self-join can be performed using INNER JOIN or LEFT JOIN, treating one instance of the table as the "left" and another as the "right."</p> <pre> SELECT country.name AS country_name, AVG(city.population) AS avg_population FROM country JOIN city ON country.code = city.countrycode GROUP BY country.name ORDER BY avg_population DESC; </pre>
Right join	<p>A RIGHT JOIN (or RIGHT OUTER JOIN) in SQL returns all records from the right table and the matching records from the left table. If there is no match, NULL values are returned for columns from the left table.</p> <pre> SELECT columns FROM left_table RIGHT JOIN right_table ON left_table.common_column = right_table.common_column; </pre>
Full join	<p>A FULL JOIN (or FULL OUTER JOIN) in SQL returns all records from both the left and right tables. If there is a match, it includes the corresponding data from both tables. If there is no match, NULL values appear for missing data from either table.</p> <pre> SELECT columns FROM left_table FULL JOIN right_table ON left_table.common_column = right_table.common_column; </pre>
Inner join	<p>An INNER JOIN in SQL returns only the matching records from both tables based on a specified condition. Rows from both tables are included only if there is a match in the common column.</p> <pre> SELECT columns FROM table1 INNER JOIN table2 ON table1.common_column = table2.common_column; </pre>

Cross join

A **CROSS JOIN** in SQL returns the **Cartesian product** of two tables, meaning it combines every row from the first table with every row from the second table. This type of join does not require a condition to match rows between the tables. As a result, the number of rows in the result will be the product of the number of rows in each table.

```
SELECT columns  
FROM table1  
CROSS JOIN table2;
```

Caution: CROSS JOIN can result in large datasets if the tables have many rows, as it produces every possible combination.

Left join

A **LEFT JOIN** (or **LEFT OUTER JOIN**) in SQL returns all records from the **left** table and the matching records from the **right** table. If there is no match, NULL values are returned for columns from the right table.

```
SELECT columns  
FROM left_table  
LEFT JOIN right_table  
ON left_table.common_column = right_table.common_column;
```



Day 4: Task 1: SQL Practical

In your groups, work together to answer the below questions. It may be of benefit if one of you shares your screen with the group and as a team answer / take screen shots from there.

Setting up the database:

1. Download world_db(1) [here](#)
2. Follow each step to create your database [here](#)

For each question I would like to see both the syntax used and the output.

1. **Count Cities in USA:** *Scenario:* You've been tasked with conducting a demographic analysis of cities in the United States. Your first step is to determine the total number of cities within the country to provide a baseline for further analysis.




```

3  select countrycode, count(name) as count_of_cities
4  from city
5  where countrycode = 'usa'
6  group by countrycode;

```

100% 22:6

Result Grid



Filter Rows:

Search

Export:



countrycode	count_of_cities
USA	274

2. **Country with Highest Life Expectancy:** *Scenario:* As part of a global health initiative, you've been assigned to identify the country with the highest life expectancy. This information will be crucial for prioritising healthcare resources and interventions.







```

8  select name, LifeExpectancy
9  from country
10 order by LifeExpectancy desc
11 limit 1;

```

00% 9:11

Result Grid   Filter Rows: Export:  Fetch rows: 

	name	LifeExpectan...
	Andorra	83.5

3. **"New Year Promotion: Featuring Cities with 'New' :** *Scenario:* In anticipation of the upcoming New Year, your travel agency is gearing up for a special promotion featuring cities with names including the word 'New'. You're tasked with swiftly compiling a list of all cities from around the world. This curated selection will be essential in creating promotional materials and enticing travellers with exciting destinations to kick off the New Year in style.

```

13 * select name
14     from city
15     where name like '%new%';

```

100% 25:15

Result Grid Filter Rows: Search Export:

name
Newcastle
Newcastle upon Tyne
Newport
Newcastle
Kowloon and New Kowloon
New Bombay
New Delhi
Khanewal
New York
New Orleans
Newark

4. **Display Columns with Limit (First 10 Rows):** *Scenario:* You're tasked with providing a brief overview of the most populous cities in the world. To keep the report concise, you're instructed to list only the first 10 cities by population from the database.

17

•

select name, Population

18

from city

19

order by Population desc

20

limit 10;

100%

↕

10:18

Result Grid

Filter Rows:

Q

Search

Exp

	name	Population	
	Mumbai (Bombay)	10500000	
	Seoul	9981619	
	São Paulo	9968485	
	Shanghai	9696300	
	Jakarta	9604900	
	Karachi	9269265	
	Istanbul	8787958	
	Ciudad de México	8591309	
	Moscow	8389200	
	New York	8008278	

5. **Cities with Population Larger than 2,000,000:** *Scenario:* A real estate developer is interested in cities with substantial population sizes for potential investment opportunities. You're tasked with identifying cities from the database with populations exceeding 2 million to focus their research efforts.

```

22  select name, Population
23  from city
24  where Population > 2000000;

```

100% 1:21

Result Grid Filter Rows: Search Export

	name	Population	
	Alger	2168000	
	Luanda	2022000	
	Buenos Aires	2982146	
	Sydney	3276207	
	Melbourne	2865329	
	Dhaka	3612850	
	São Paulo	9968485	
	Rio de Janeiro	5598953	
	Salvador	2302832	
	Belo Horizonte	2139125	
	Fortaleza	2097757	

6. **Cities Beginning with 'Be' Prefix:** *Scenario:* A travel blogger is planning a series of articles featuring cities with unique names. You're tasked with compiling a list of cities from the database that start with the prefix 'Be' to assist in the blogger's content creation process.

```

26 • select name, Population
27     from city
28     where name like 'b%';

```

00% 22:28

Result Grid Filter Rows: Search

	name	Population	
	Breda	160398	
	Batna	183377	
	Biskra	128281	
	Blida (el-Boulaida)	127284	
	Béjaïa	117162	
	Béchar	107311	
	Benguela	128300	
	Buenos Aires	2982146	
	Berazategui	276916	
	Bahía Blanca	239810	
	Brisbane	1291117	
	B...	1707000	

city 1

- Cities with Population Between 500,000-1,000,000:** *Scenario:* An urban planning committee needs to identify mid-sized cities suitable for infrastructure development projects. You're tasked with identifying cities with populations ranging between 500,000 and 1 million to inform their decision-making process.

29

30

31

32

33

```

select name, Population
from city
where Population between 500000 and 1000000
order by name;

```

100%

1:29

Result Grid

Filter Rows:

Search

Export:

	name	Population	
	Acapulco de Juárez	721011	
	Adelaide	978100	
	Agra	891790	
	Aguascalientes	643360	
	Ahvaz	804980	
	Allahabad	792858	
	Almirante Brown	538918	
	Amman	1000000	
	Amoy [Xiamen]	627500	
	Amritsar	708835	
	Amsterdam	731200	
	Amstelveen	510011	

8. **Display Cities Sorted by Name in Ascending Order:** *Scenario:* A geography teacher is preparing a lesson on alphabetical order using city names. You're tasked with providing a sorted list of cities from the database in ascending order by name to support the lesson plan.

35	select name
36	from city
37	order by name;
00%	10:36
Result Grid	
Filter Rows: Search	
name	
[San Cristóbal de] la Laguna	
's-Hertogenbosch	
A Coruña (La Coruña)	
Aachen	
Aalborg	
Aba	
Abadan	
Abaetetuba	
Abakan	
Abbotsford	
Abeokuta	
Result 4	country 5
	city 6

9. **Most Populated City:** *Scenario:* A real estate investment firm is interested in cities with significant population densities for potential development projects. You're tasked with identifying the most populated city from the database to guide their investment decisions and strategic planning.


```

39  select name, Population
40  from city
41  order by Population desc
42  limit 1;

```

00% 9:42

Result Grid Filter Rows: Search

name	Population
Mumbai (Bombay)	10500000

city 12

10. City Name Frequency Analysis: Supporting Geography Education

Scenario: In a geography class, students are learning about the distribution of city names around the world. The teacher, in preparation for a lesson on city name frequencies, wants to provide students with a list of unique city names sorted alphabetically, along with their respective counts of occurrences in the database. You're tasked with this sorted list to support the geography teacher.

```

43
44 select name, count(*)
45 from city
46 group by name
47 order by count(*) desc;

```

00% 1:43

Result Grid Filter Rows: Search Expo

name	count(*)
San José	4
Córdoba	3
San Miguel	3
San Fernando	3
Hamilton	3
La Paz	3
Toledo	3
Cambridge	3
Springfield	3
Richmond	3
Valencia	3

Result 15

11.City with the Lowest Population: *Scenario:* A census bureau is conducting an analysis of urban population distribution. You're tasked with identifying the city with the lowest population from the database to provide a comprehensive overview of demographic trends.

```

49 select name, Population
50 from city
51 order by Population
52 limit 1;

```

00% 1:49

Result Grid Filter Rows: Search

name	Population
Adamstown	42

city 16

12. Country with Largest Population: *Scenario:* A global economic research institute requires data on countries with the largest populations for a comprehensive analysis. You're tasked with identifying the country with the highest population from the database to provide valuable insights into demographic trends.

```

54 select name, Population
55 from country
56 order by Population desc
57 limit 1;

```

0% 1:54

Result Grid Filter Rows: Search

name	Population
China	1277558000

country 18

13.Capital of Spain: *Scenario:* A travel agency is organising tours across Europe and needs accurate information on capital cities. You're tasked with identifying the capital of Spain from the database to ensure itinerary accuracy and provide travellers with essential destination information.

```

59 SELECT city.name AS capital
60 FROM country
61 JOIN city ON country.capital = city.id
62 WHERE country.name = 'Spain';

```

00% 30:62

Result Grid Filter Rows: Search Export:

capital
Madrid

Result 19

14. Cities in Europe: *Scenario:* A European cultural exchange program is seeking to connect students with cities across the continent. You're tasked with compiling a list of cities located in Europe from the database to facilitate program planning and student engagement.

```

63
64 select country.name, city.name
65 from country
66 join city
67 on country.code = city.CountryCode
68 where country.Region like '%europe%';

```

00% 31:64

Result Grid Filter Rows: Search Export:

	name	name
	Albania	Tirana
	Andorra	Andorra la Vella
	Austria	Wien
	Austria	Graz
	Austria	Linz
	Austria	Salzburg
	Austria	Innsbruck
	Austria	Klagenfurt
	Belgium	Antwerpen
	Belgium	Gent
	Belgium	Charleroi

Result 21

15.Average Population by Country: *Scenario:* A demographic research team is conducting a comparative analysis of population distributions across countries. You're tasked with calculating the average population for each country from the database to provide valuable insights into global population trends.

74	SELECT
75	country.name AS country_name,
76	AVG(city.population) AS avg_population
77	FROM country
78	JOIN city ON country.code = city.countrycode
79	GROUP BY country.name
80	ORDER BY avg_population DESC;

00%	1:74
-----	------

Result Grid	Filter Rows:	Search	Export:
-------------	--------------	--------	---------

country_name	avg_populati...
Singapore	4017733.0000
Hong Kong	1650316.5000
Uruguay	1236000.0000
Guinea	1090610.0000
Uganda	890800.0000
Liberia	850000.0000
Sierra Leone	850000.0000
Mali	809552.0000
Australia	808119.0000
Mongolia	773700.0000
Congo	725000.0000

Result 23

16.Capital Cities Population Comparison: *Scenario:* A statistical analysis firm is examining population distributions between capital cities worldwide. You're tasked with comparing the populations of capital cities from different countries to identify trends and patterns in urban demographics.

```

81
82 • select city.name, country.population
83   from city
84   join country
85   on city.id = country.capital
86   order by city.name;

```

00% 20:86

Result Grid Filter Rows: Export:

	name	population	
	Abu Dhabi	2441000	
	Abuja	111506000	
	Accra	20212000	
	Adamstown	50	
	Addis Abeba	62565000	
	Agaña	168000	
	al-Manama	617000	
	Alger	31471000	
	Alofi	2000	
	Amman	5083000	
	Amsterdam	15864000	
	Andorra la...	78000	

Result 4

17. Countries with Low Population Density: *Scenario:* An agricultural research institute is studying countries with low population densities for potential agricultural development projects. You're tasked with identifying countries with sparse populations from the database to support the institute's research efforts.

88	SELECT
89	name AS country_name,
90	population,
91	surfacearea,
92	(population / surfacearea) AS population_density
93	FROM country
94	WHERE population > 0 AND surfacearea > 0 -- Avoid division by zero
95	ORDER BY population_density ASC;

00%	33:95
-----	-------

Result Grid	Filter Rows:	Search	Export:
-------------	--------------	--------	---------

country_name	population	surfacearea	population_dens...
Greenland	56000	2166090.00	0.0259
Svalbard and Jan Mayen	3200	62422.00	0.0513
Falkland Islands	2000	12173.00	0.1643
Pitcairn	50	49.00	1.0204
Western Sahara	293000	266000.00	1.1015
Mongolia	2662000	1566500.00	1.6993
French Guiana	181000	90000.00	2.0111
Namibia	1726000	824292.00	2.0939
Australia	18886000	7741220.00	2.4397
Suriname	417000	163265.00	2.5541
Mauritania	2670000	1025520.00	2.6036
Iceland	279000	103000.00	2.7087

Result 5	country 6	city 7	city 8	city 9	city 10
----------	-----------	--------	--------	--------	---------

18.Cities with High GDP per Capita: *Scenario:* An economic consulting firm is analysing cities with high GDP per capita for investment opportunities. You're tasked with identifying cities with above-average GDP per capita from the database to assist the firm in identifying potential investment destinations.

```

103 SELECT
104     city.name AS city_name,
105     country.name AS country_name,
106     (country.GNP / country.population) AS gdp_per_capita
107 FROM city
108 JOIN country ON city.countrycode = country.code
109 WHERE country.GNP IS NOT NULL AND country.population > 0 -- Avoid division by zero
110 HAVING gdp_per_capita > (SELECT AVG(GNP / population) FROM country WHERE GNP IS NOT NULL AND population > 0)
111 ORDER BY gdp_per_capita DESC
112 LIMIT 10;

```

100% 12:104

Result Grid Filter Rows: Search Export: Fetch rows:

city_name	country_name	gdp_per_capi...
Luxembourg [Luxemburg/Lëtzebuerg]	Luxembourg	0.037459
Zürich	Switzerland	0.036936
Geneve	Switzerland	0.036936
Basel	Switzerland	0.036936
Bern	Switzerland	0.036936
Lausanne	Switzerland	0.036936
Saint George	Bermuda	0.035815
Hamilton	Bermuda	0.035815
Bandar Seri Begawan	Brunei	0.035686
Schaan	Liechtenstein	0.034644

Result 25

19. Display Columns with Limit (Rows 31-40): *Scenario:* A market research firm requires detailed information on cities beyond the top rankings for a comprehensive analysis. You're tasked with providing data on cities ranked between 31st and 40th by population to ensure a thorough understanding of urban demographics.



```
113
114 SELECT
115     name AS city_name,
116     countrycode,
117     population
118 FROM city
119 ORDER BY population DESC
120 LIMIT 10
121 OFFSET 30;
```

100% 1:114

Result Grid Filter Rows: Search Export: Fetch rows:

	city_name	countrycode	population
	Shenyang	CHN	4265200
	Kanton [Guangzhou]	CHN	4256300
	Singapore	SGP	4017733
	Ho Chi Minh City	VNM	3980000
	Chennai (Madras)	IND	3841396
	Pusan	KOR	3804522
	Los Angeles	USA	3694820
	Dhaka	BGD	3612850
	Berlin	DEU	3386667
	Rangoon (Yangon)	MMR	3361700

city 26

Day 4: Task 2: Written (Optional)

In your groups, discuss and complete the below activity. You can either nominate one writer or split the elements between you. Everyone however must have the completed work below:

Imagine you have been hired by a small retail business that wants to streamline its operations by creating a new database system. This database will be used to manage inventory, sales, and customer information. The business is a small corner shop that sells a range of groceries and domestic products. It might help to picture your local convenience store and think of what they sell. They also have a loyalty program, which you will need to consider when deciding what tables to create.

Write a 500-word essay explaining the steps you would take to set up and create this database. Your essay should cover the following points:

1. **Understanding the Business Requirements:**
 - a. What kind of data will the database need to store?
 - b. Who will be the users of the database, and what will they need to accomplish?
2. **Designing the Database Schema:**
 - a. How would you structure the database tables to efficiently store inventory, sales, and customer information?
 - b. What relationships between tables are necessary (e.g., how sales relate to inventory and customers)?
3. **Implementing the Database:**
 - a. What SQL commands would you use to create the database and its tables?
 - b. Provide examples of SQL statements for creating tables and defining relationships between them.
4. **Populating the Database:**
 - a. How would you input initial data into the database? Give examples of SQL INSERT statements.
5. **Maintaining the Database:**
 - a. What measures would you take to ensure the database remains accurate and up to date?
 - b. How would you handle backups and data security?

Your essay should include specific examples of SQL commands and explain why each step is necessary for creating a functional and efficient database for the retail business.

Please write
your 500-
word essay
here

Designing and Implementing a Database for a Small Retail Business

Understanding the Business Requirements

To create an efficient database for a small retail business, it is crucial to understand the data that needs to be stored and the users who will interact with it. The database should include information on:

- **Inventory:** Product names, categories, stock levels, supplier details, and pricing.



- **Sales:** Transaction records, date and time of sales, items sold, total amounts, and payment methods.
- **Customers:** Personal details (name, phone number, email), purchase history, and loyalty program details.
- **Employees:** Staff managing sales and inventory, along with user roles and permissions.

The primary users of the database will be store employees, managers, and possibly suppliers. Employees will need to process sales, update inventory, and register customers in the loyalty program. Managers will require access to reports on sales trends, stock levels, and customer activity.

Designing the Database Schema

A well-structured schema ensures data integrity and efficient retrieval. The key tables and their relationships include:

- **Products (Inventory Management):** Stores product details.
- **Customers:** Tracks customer information.
- **Sales:** Records each transaction, linking to products and customers.
- **Sales_Items:** Manages the many-to-many relationship between sales and products.
- **Loyalty_Program:** Tracks points and rewards for customers.

Table Relationships

- A **sale** belongs to one **customer**, but a customer can have multiple sales (one-to-many).
- A **sale** consists of multiple **products**, and a product can be in multiple sales (many-to-many, requiring a junction table **Sales_Items**).
- The **loyalty program** links to the **customers** table.

Implementing the Database

Using SQL, we create the tables with appropriate relationships:

```
CREATE DATABASE RetailDB;
USE RetailDB;
```

```
CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    category VARCHAR(50),
    price DECIMAL(10,2) NOT NULL,
    stock_quantity INT NOT NULL,
    supplier VARCHAR(100)
);
```

```
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    phone VARCHAR(15),
    email VARCHAR(100),
    loyalty_points INT DEFAULT 0
```

```
);
```

```
CREATE TABLE Sales (  
    sale_id INT PRIMARY KEY AUTO_INCREMENT,  
    customer_id INT,  
    sale_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
    total_amount DECIMAL(10,2),  
    FOREIGN KEY (customer_id) REFERENCES  
Customers(customer_id)  
);
```

```
CREATE TABLE Sales_Items (  
    sale_id INT,  
    product_id INT,  
    quantity INT,  
    subtotal DECIMAL(10,2),  
    PRIMARY KEY (sale_id, product_id),  
    FOREIGN KEY (sale_id) REFERENCES Sales(sale_id),  
    FOREIGN KEY (product_id) REFERENCES Products(product_id)  
);
```

Populating the Database

Initial data can be inserted using SQL:

```
INSERT INTO Products (name, category, price, stock_quantity,  
supplier) VALUES  
( 'Milk', 'Dairy', 1.50, 100, 'Local Dairy Co.' ),  
( 'Bread', 'Bakery', 2.00, 50, 'Bakery Supplies Ltd.' );
```

```
INSERT INTO Customers (name, phone, email) VALUES  
( 'John Doe', '1234567890', 'john@example.com' );
```

```
INSERT INTO Sales (customer_id, total_amount) VALUES  
(1, 15.00);
```

```
INSERT INTO Sales_Items (sale_id, product_id, quantity,  
subtotal) VALUES  
(1, 1, 2, 3.00),  
(1, 2, 1, 2.00);
```

Maintaining the Database

To ensure data accuracy and security:

- **Data Validation:** Implement constraints (e.g., NOT NULL, CHECK) to prevent incorrect data entries.



- **Regular Backups:** Schedule daily or weekly backups using MySQL's `mysqldump` command: `mysqldump -u root -p RetailDB > backup.sql`
-
- **Data Security:** Assign roles and permissions using SQL's `GRANT` statement: `GRANT SELECT, INSERT, UPDATE ON RetailDB.* TO 'store_user'@'localhost';`
-

By following these steps, the small retail business can effectively manage its operations, ensuring accurate inventory tracking, streamlined sales processing, and customer relationship management. This database will help improve efficiency, reduce errors, and support future growth.



Course Notes

It is recommended to take notes from the course, use the space below to do so, or use the revision guide shared with the class:



We have included a range of additional links to further resources and information that you may find useful, these can be found within your revision guide.

END OF WORKBOOK

Please check through your work thoroughly before submitting and update the table of contents if required.

Please send your completed work booklet to your trainer.

