Just IT

B2W group

Apprenticeships | Training | Recruitment

Data Technician

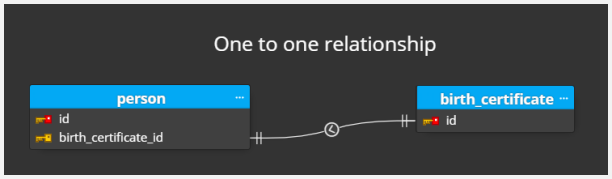
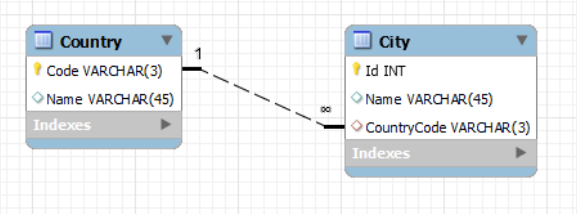
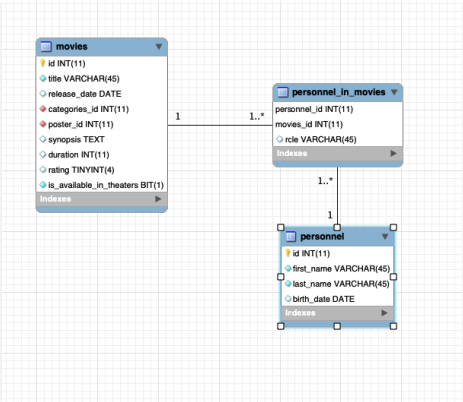
Name: Sara Haider

Course Date: 01/05/2025

Table of contents

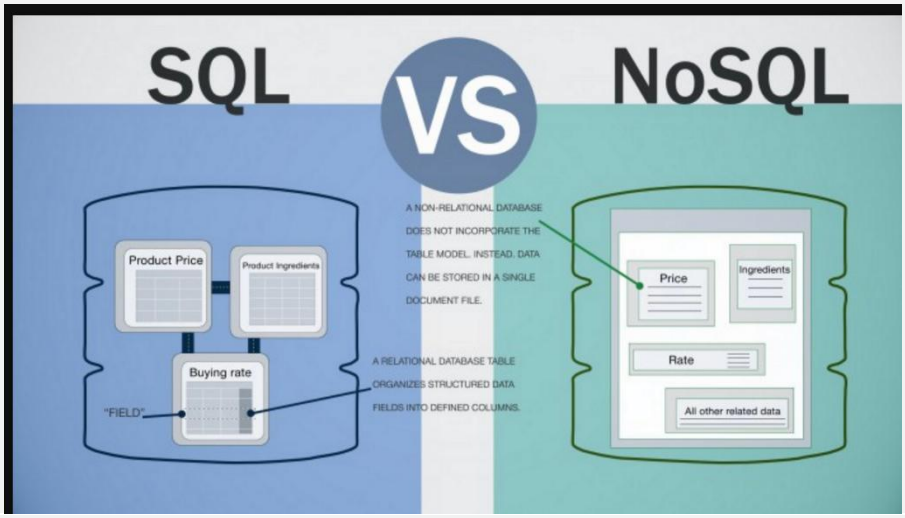
Day 1: Task 1

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

What is a primary key?	A primary key uniquely identifies each record in a table.
How does this differ from a secondary key?	A secondary key does not have to be unique and doesn't serve as the primary way to uniquely identify a record. This can allow NULL values whereas every record must have a value for the primary key.
How are primary and foreign keys related?	A foreign key is a field in one table that links to the primary key of another table. This allows you to join the data from the two tables based on a common attribute.
Provide a real-world example of a one-to-one relationship	Each person has one birth certificate and each birth certificate is associated with one person. 
Provide a real-world example of a one-to-many relationship	A Country can have many cities but each city is located in one country. 
Provide a real-world example of a many-to-many relationship	Movies can feature multiple actors and actors can appear in multiple movies. 

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?	A relational database stores data in a structured way. They require a schema for the data where you define where information goes. Nonrelational databases don't use tables and rows, data is stored in more flexible formats e.g. documents, key-value. Relational is better when you need order and structure whereas non-relational is better when flexibility and scalability is needed. 
What type of data would benefit off the non-relational model? **Why?**	Non-relational databases are ideal for handling large, flexible or unstructured data like social media posts, images, which don't fit into tables. They are great for real-time data and can handle big data that needs to grow quickly. These databases work well when the structure of the data changes my product list all user profiles. They can also handle complex data like orders with multiple items. They are built to stay online even if something goes wrong which ensures minimal downtime.

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	Where a table is joined to itself, compares rows within the same table. So gives you only the records that match in both tables. Example: Show customers who share the same last name.
Right join	This returns all rows from the right table, and matched rows from the left table. NULLs appear for unmatched left-side entries. Example: show all orders even if customer details are not complete.
Full join	Returns all rows from both tables. If something does not match on one side, it still shows up with NULLs for the missing info. Example: combine all customers and all orders whether or not each customer has ordered or each order has a known customer.
Inner join	Gives only the records that match in both tables. Example: Show customers who have placed at least one order. Customers customer_id, first_name, last_name, age, country 1, John, Doe, 31, USA 2, Robert, Luna, 22, USA 3, David, Robinson, 22, UK 4, John, Reinhardt, 25, UK 5, Betty, Doe, 28, UAE Orders order_id, item, amount, customer_id 1, Keyboard, 400, 4 2, Mouse, 300, 4 3, Monitor, 12000, 3 4, Keyboard, 400, 1 5, Mousepad, 250, 2
Cross join	Combines every row from the first table with every row from the second table.

	Example: generate every combination of customers and products for a survey or promotion.
Left join	Shows all records from the first left table and adds matching info from the second table. If there is no match, it shows NULL. Example: Show all customers and include their orders if they have any.

Day 4: Task 1: Written

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:

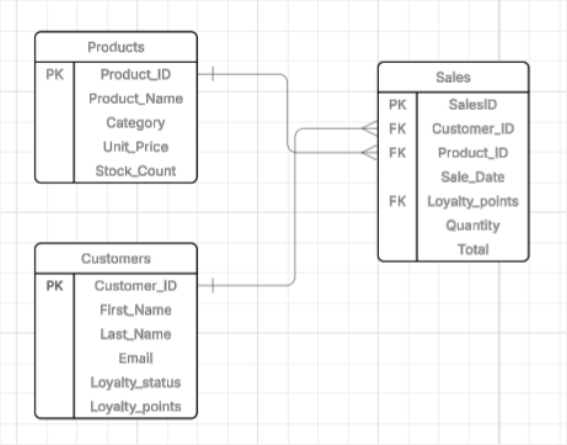
1. ***Understanding the Business Requirements**:*
    a. *What kind of data will the database need to store?*
    b. *Who will be the users of the database, and what will they need to accomplish?*
2. ***Designing the Database Schema**:*
    a. *How would you structure the database tables to efficiently 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

1. The first step in creating a database system is understanding business needs. For a small retail store, the database would cover products (inventory), sales, and customer details.
The products table tracks inventory, including quantities and prices, helping the shop manager monitor stock and sales trends. The sales table records transactions, such as date, products, and quantities sold, ensuring the inventory is updated after each sale.
The customer table stores information like names, contact details, and loyalty points, allowing staff to track and update points during transactions.
To keep the system running smoothly, the IT support team must regularly maintain the database.

2.



The entity-relationship diagram shows three tables: Products, Customers, and Sales. Each Product has a unique ID and includes details like name, price, and stock. Each Customer is identified by a Customer_ID with personal info, email, and loyalty points. The Sales table tracks transactions, using a SalesID as the primary key, with foreign keys linking to both Customer_ID and Product_ID. It includes sale date, quantity, total cost, and loyalty points earned or used. The relationships are one-to-many: a customer can have multiple sales, and a product can appear in many sales.

3. The SQL commands used include CREATE DATABASE to create the database. CREATE TABLE was used to create the 3 tables, Products, Sales and Customers. In these tables the fields are entered including their data types to identify verify inputs.

```sql
CREATE DATABASE day_4_task_1;
USE day_4_task_1;
CREATE TABLE Products (
    Product_ID INT PRIMARY KEY,
    Product_Name VARCHAR(50),
    Category VARCHAR(50),
    Unit_Price DECIMAL(10,2),
    Stock_Count INT
);

CREATE TABLE Sales (
    Sales_ID INT PRIMARY KEY,
    Customer_ID INT,
    Product_ID INT,
    Sale_Date DATE,
    Loyalty_points INT,
    Quantity INT,
    Total DECIMAL(10,2),
    FOREIGN KEY (Customer_ID) REFERENCES Customers(Customer_ID),
    FOREIGN KEY (Product_ID) REFERENCES Products(Product_ID)
);

CREATE TABLE Customers (
    Customer_ID INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Email VARCHAR(100),
    Loyalty_points INT,
    Loyalty_status VARCHAR(100)
);
```

4. Once the database is set up, data is added using the INSE
RT INTO function. Products are added with the product ID, name, category, price and stock availability; customers with customer ID name e-mail loyalty points and if they are in the loyalty programme; sales with the sales id, customer id, product id, the sale date, loyalty points, the quantity of products, and the total of sales.

```sql
INSERT INTO Products (Product_ID, Product_Name, Category, Unit_Price, Stock_Count)
VALUES (1,'Toothpaste','Health',1.50,50);

INSERT INTO Customers (Customer_ID, First_Name, Last_Name, Email, Loyalty_Points, Loyalty_Status)
VALUES (1,'Jane','Smith','jane.smith@hotmail.com',100,'yes');

INSERT INTO Sales (Sales_ID, Customer_ID, Product_ID, Sale_Date, Loyalty_Points, Quantity, Total)
VALUES (1,1,1,'2025-01-15',100,3,1000.00);
```

```
SELECT * FROM Products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

Product_ID	Product_Name	Category	Unit_Price	Stock_Count
1	Toothpaste	Health	1.50	50

```
SELECT * FROM Customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_ID	First_Name	Last_Name	Email	Loyalty_points	Loyalty_status
1	Jane	Smith	jane.smith@hotmail.com	100	yes

```
SELECT * FROM Sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Sales_ID	Customer_ID	Product_ID	Sale_Date	Loyalty_points	Quantity	Total
1	1	1	2025-01-15	100	3	1000.00

5. To Ensure data accuracy enforcing validation rules at both application and database levels is necessary. Including using data types, constraints, and foreign keys to prevent errors.

The screenshot on the right shows a query being ran to check if there are any missing product names, the result came back empty.

```
SELECT *
FROM Products
WHERE Product_Name IS NULL;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

Product_ID	Product_Name	Category	Unit_Price	Stock_Count

```
SELECT Customer_ID, First_Name, Last_Name, Loyalty_points, Loyalty_Status
FROM Customers
WHERE First_Name = 'Jane' AND Last_Name = 'Smith';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_ID	First_Name	Last_Name	Loyalty_points	Loyalty_Status
1	Jane	Smith	100	yes

The screenshot shows a query being ran searching for a specific client, the purpose is to check their status in the loyalty program, if they are involved or not and their count.

The screenshot shows a query looking for duplications, this example shows numerous were found for the following records, proving this data must be cleaned.

```
SELECT First_Name, Last_Name, Email, COUNT(*) AS DuplicateCount
FROM Customers
GROUP BY First_Name, Last_Name, Email
HAVING COUNT(*) > 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

First_Name	Last_Name	Email	DuplicateCount
Jane	Smith	jane.smith@hotmail.com	9
Renzo	Ali	renzo.ali@hotmail.com	4

Backups:
Protect data with a strategy that includes daily full backups and more frequent differential or log backups, based on data change rates.

Access Control:
Use role-based permissions (e.g., Admin, Analyst, AppUser) to simplify management and apply the least privilege principle.

Day 4: Task 2: 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)**
2. **Follow each step to create your database**

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.

```
SELECT count(countryCode) as City_count, CountryCode FROM city
WHERE CountryCode= 'USA';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

City_count	CountryCode
274	USA

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.

```
SELECT country.name
FROM country
ORDER BY country.lifeexpectancy DESC
limit 1;
```

Result Grid | Filter Rows: | Export:

name
Andorra

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.

```
SELECT * FROM city WHERE name like 'New%';
```

Result Grid | Filter Rows: | Edit: | Export/In

ID	Name	CountryCode	District	Population
137	Newcastle	AUS	New South Wales	270324
482	Newcastle upon Tyne	GBR	England	189150
502	Newport	GBR	Wales	139000
734	Newcastle	ZAF	KwaZulu-Natal	222993
1106	New Bombay	IND	Maharashtra	307297
1109	New Delhi	IND	Delhi	301297
3793	New York	USA	New York	8008278
3823	New Orleans	USA	Louisiana	484674
3855	Newark	USA	New Jersey	273546
3905	Newport News	USA	Virginia	180150
3971	New Haven	USA	Connecticut	123626

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.

```
SELECT city.Name
FROM city
ORDER BY city.Population DESC
LIMIT 10;
```

Result Grid | Filter Rows:

Name
Mumbai (Bombay)
Seoul
São Paulo
Shanghai
Jakarta
Karachi
Istanbul
Ciudad de México
Moscow
New York

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.

```
SELECT * FROM city WHERE name like 'New%';

SELECT NAME,Population FROM city WHERE Population>2000000;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetc

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

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.

```
SELECT DISTINCT * FROM city WHERE name like 'Be%';
```

Result Grid | Filter Rows: | Edit: | Export

ID	Name	CountryCode	District	Population
45	Béjaïa	DZA	Béjaïa	117162
49	Béchar	DZA	Béchar	107311
59	Benguela	AGO	Benguela	128300
93	Berazategui	ARG	Buenos Aires	276916
184	Belize City	BLZ	Belize City	55810
185	Belmopan	BLZ	Cayo	7105
209	Belo Horizonte	BRA	Minas Gerais	2139125
216	Belém	BRA	Pará	1186926
246	Belford Roxo	BRA	Rio de Janeiro	425194
266	Betim	BRA	Minas Gerais	302108

7. **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.

```
SELECT Name, Population
FROM city
WHERE Population BETWEEN 500000 AND 1000000;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

Name	Population
Amsterdam	731200
Rotterdam	593321
Oran	609823
Dubai	669181
Rosario	907718
Lomas de Zamora	622013
Quilmes	559249
Almirante Brown	538918
La Plata	521936
Mar del Plata	512880
Adelaide	978100

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.

```
SELECT Name, Population
FROM city
ORDER BY Name ASC;
```

Result Grid | Filter Rows: | Export

Name	Population
[San Cristóbal de] la Laguna	127945
´s-Hertogenbosch	129170
A Coruña (La Coruña)	243402
Aachen	243825
Aalborg	161161
Aba	298900
Abadan	206073
Abaetetuba	111258
Abakan	169200
Abbotsford	105403

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.

```
SELECT Name, Population
FROM city
ORDER BY Population DESC
LIMIT 10;
```

Result Grid | Filter Rows:

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

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.

```
SELECT DISTINCT name, count(name) as No_of_occurance FROM city GROUP BY name ORDER BY name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	No_of_occurance
[San Cristóbal de] la Laguna	1
´s-Hertogenbosch	1
A Coruña (La Coruña)	1
Aachen	1
Aalborg	1
Aba	1
Abadan	1
Abaetetuba	1
Abakan	1
Abbotsford	1
Abeokuta	1
Aberdeen	1
Abha	1
Abidjan	1

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.

```
SELECT city.name, city.population
FROM city
ORDER BY population ASC
LIMIT 1;
```

Result Grid | Filter Rows: | Exp

name	population
Adamstown	42

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.

```
SELECT country.name, country.population
FROM country
ORDER BY population ASC
LIMIT 1;
```

Result Grid | Filter Rows: | Export:

name	population
Antarctica	0

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.

```
SELECT Name
FROM city
WHERE ID =
(SELECT Capital
FROM country
WHERE Name = 'Spain'
);
```

Result Grid | Filter Rows:

Name
Madrid

14. **Country with Highest Life Expectancy:** *Scenario:* A healthcare foundation is conducting research on global health indicators. You're tasked with identifying the country with the highest life expectancy from the database to inform their efforts in improving healthcare systems and policies.

```
SELECT country.name
FROM country
ORDER BY country.lifeexpectancy DESC
limit 1;
```

Result Grid | Filter Rows: | Export:

name
Andorra

15. **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.

```
SELECT city.Name, country.Name
FROM city
JOIN Country on city.CountryCode = country.code
WHERE country.Continent = 'Europe';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

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

16. **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.

```
SELECT Name, AVG(Population) Average_Population
FROM country
GROUP BY Name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

Name	Average_Population
Aruba	103000.0000
Afghanistan	22720000.0000
Angola	12878000.0000
Anguilla	8000.0000
Albania	3401200.0000
Andorra	78000.0000
Netherlands Antilles	217000.0000
United Arab Emirates	2441000.0000

17. **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.

```
SELECT country.Name Country, city.Name Capital_City, city.Population
FROM country
JOIN city ON country.Capital= city.ID
ORDER BY city.Population DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Country	Capital_City	Population
South Korea	Seoul	9981619
Indonesia	Jakarta	9604900
Mexico	Ciudad de México	8591309
Russian Federation	Moscow	8389200
Japan	Tokyo	7980230
China	Peking	7472000
United Kingdom	London	7285000

18. **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.

```
SELECT country.name, country.population
FROM country
ORDER BY population ASC
LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wra

name	population
South Georgia and the South Sandwich Islands	0
French Southern territories	0
Bouvet Island	0
British Indian Ocean Territory	0
Antarctica	0
United States Minor Outlying Islands	0
Heard Island and McDonald Islands	0
Pitcairn	50
Cocos (Keeling) Islands	600
Holy See (Vatican City State)	1000

19. **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.

```
SELECT city.Name, country.GNP/country.Population GDP
FROM country
INNER JOIN city ON country.code = city.countryCode
ORDER BY GDP DESC
;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

Name	GDP
Luxembourg [Luxemburg/Lëtzebuerg]	0.037459
Zürich	0.036936
Geneve	0.036936
Basel	0.036936
Bern	0.036936
Lausanne	0.036936
Saint George	0.035815
Hamilton	0.035815
Bandar Seri Begawan	0.035686
Schaan	0.034644

20. **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.

```
SELECT city.name, city.population
FROM city
ORDER BY population DESC
LIMIT 10 OFFSET 30;
```

Result Grid | Filter Rows: | Export:

name	population
Shenyang	4265200
Kanton [Guangzhou]	4256300
Singapore	4017733
Ho Chi Minh City	3980000
Chennai (Madras)	3841396
Pusan	3804522
Los Angeles	3694820
Dhaka	3612850
Berlin	3386667
Rangoon (Yangon)	3361700

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:

Databases
Used to store and organise information.
Entities
Customer (custID, title, firstname, surname, email)
Guide or product (productID, subject, level, price)
Subscription(subID, dtartDate, endDate)

Entity identifier
The identifier is known as the primary key. This uniquely identifies a particular record.
PK- Primary Key
FK- Foreign key

Relationship = cardinality
Entity relationship = schema

Schema is a visual representation of relationships between entities
3 types of schemas
Conceptual
Star
Snowflake

Star schema
Centred fact table surrounding dimension tables
many to one relationship

Relational database
A database that stores structured data

SQL
Structured query language
CRUD= create, read, update, delete
Used for storing, manipulating and retrieving data from relational databases.
Warehouse is meant for analysis

Data types
String- char (0-255), varchar
Numeric- INT, Decimal
Date&Time- DateTime, Date (YY-MM-DD), Time (HH:MM:SS)

SELECT- Retrieve data
*=wildcard

```
SELECT *
FROM Table
;
```

%a means ends with a

breakdown means aggregate function

```
SELECT first_name, last_name, age
FROM customers
WHERE country IN ('UK' , 'USA') AND age>22
;
```

Altering tables
Creating, adding, dropping columns

Subquery
A query within a query

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.