



Just IT

 B2Wgroup

Apprenticeships | Training | Recruitment

Data Technician

Name:

Course Date:

Table of contents

Day 1: Task 1 3

Day 1: Task 2 4

Day 3: Task 1 4

Day 4: Task 1: Written 5

Day 4: Task 2: SQL Practical 9

Course Notes24

Additional Information.....25



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 is usually a scaling integer that indicates a row of data.
How does this differ from a secondary key?	A secondary key would be for example a phone number or an email, something that allows a record's data retrieval.
How are primary and foreign keys related?	Foreign keys on a table A will refer, have a relationship with a primary key in table B.
Provide a real-world example of a one-to-one relationship	For example, in a school a student id is assigned to 1 student, and a student will only be assigned 1 student id
Provide a real-world example of a one-to-many relationship	A customer may have many orders, so 1 customer may have many order id's in his row of data.
Provide a real-world example of a many-to-many relationship	Many students may enrol in many courses, just as many courses may allow many students to enrol.



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 utilises structured tables, meaning that data added to tables must follow a certain order and data type. A non-relational database utilises NO-SQL, meaning that it stores data in a non-tabular format, therefore not requiring any structure when adding rows to a table. Non-relational databases require you to use different languages, for example JSON and are a lot easier to access through APIs.
What type of data would benefit off the non-relational model?	Large amounts of unstructured data, for example an Instagram feed, which contains captions, images, videos, likes, comments, shares. This is because having structured will have null values for every post made, since not every post may contain a caption, images or videos. Therefore unstructured data will make it easier and more efficient since the table will not require to have all values for each row.
Why?	

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	A self-join table allows the table to join to itself, which enables the comparison of rows within the same table. This can be used by HR to report structured hierarchies.
Right join	A right-join will pull records from the right table and the matching records from the left table. If there is no match a NULL value will appear in the left table column. This can be used to find out what student has not enrolled in a course
Full join	A full-join pulls records from both right and left tables. This can be used to check all records against each other.
Inner join	An Inner-join creates a table that displays matching data from both left and right tables. This may be used to find out what customers placed an order.
Cross join	A cross-join creates a table with every possible combination between 2 tables. This may be used for an in-depth analysis.
Left join	A left-join returns records from the left table, as well as matching records from the right table.



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? encryption
 - 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.



- 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?*

For this retail business I will have to consider what kind of data I will need to store, this will include: stock, supply, customers and sales. I considered introducing another table for employees, but seeing how the business is small I decided against it since the number of employees will not be above 5.

The intended user will be the employee at the counter or the employee in charge of the stock.

For this database I will use a relational scheme, with the customer and inventory tables being primary tables. The Inbound(supply) and the Sales tables will link to ProductID from the Inventory(stock) table as well as the Sales table also having a link to CustomerID from the Customer table.

To create the database, I will use the "CREATE DATABASE" command

```
CREATE DATABASE cornershop;  
  
USE cornershop;
```

, as for the tables I will use the "CREATE TABLE"

```
CREATE TABLE inventory (  
    ProductID INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    product_description VARCHAR(50),  
    category VARCHAR(15),  
    pack_size INT,  
    price FLOAT,  
    inv_quantity INT  
);  
  
CREATE TABLE inbound(  
    delivery_date DATE NOT NULL,  
    ProductID INT NOT NULL,  
    quantity INT NOT NULL,  
    FOREIGN KEY (ProductID) REFERENCES inventory(ProductID)  
);
```

command

```

• CREATE TABLE customer(
  CustomerID INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  first_name VARCHAR(20),
  last_name VARCHAR(20),
  phone_number VARCHAR(11),
  email LONG VARCHAR,
  loyalty_scheme BOOLEAN,
  loyalty_points INT
);

• CREATE TABLE sales(
  OrderID INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  sale_date date,
  ProductID INT,
  quantity INT,
  CustomerID INT,
  FOREIGN KEY (CustomerID) REFERENCES customer(CustomerID),
  FOREIGN KEY (ProductID) REFERENCES inventory(ProductID)
);

```

In the screenshots above you can observe how the command “FOREIGN KEY” is used while creating the tables. This established cardinality between tables. The sales table will have a “Many to One” cardinality with the customer table, and a “Many to Many” cardinality with the product table, as many sales can be done by 1 customer, but Many sales could have Many products.

The Inbound table will have a “Many to One” cardinality with the Inventory table, as 1 product may have many deliveries.

To populate the tables I will make the use of the “INSERT INTO” command. Some examples of the SQL statement could look like this:

- **INSERT INTO**
inventory(product_description, category, pack_size, price, inv_quantity)
VALUES
('onions', 'fresh produce', 6, 1, 3),
('pepsi', 'drinks', 6, 3, 1),
('opal fruits', 'sweets', 1, 0.5, 0);
- **INSERT INTO**
inbound(delivery_date, ProductID, quantity)
VALUES
('2025-01-01', 1, 3),
('2025-01-02', 2, 10),
('2025-01-03', 1, 6);
- **INSERT INTO**
customer(first_name, last_name, phone_number, email, loyalty_scheme, loyalty_point)
VALUES
('Andrew', 'Salmon', '1111111111', 'as@yes.co.uk', TRUE, 3),
('Rohit', 'Sharma', '2222222222', 'rs@yes.co.uk', FALSE, NULL),
('Gabby', 'Kow', '3333333333', 'gk@yes.co.uk', TRUE, 5);
- **INSERT INTO**
sales(sale_date, ProductID, quantity, CustomerID)
VALUES
('2025-02-19', 1, 1, 1),
('2025-02-20', 2, 2, 2),
('2025-02-20', 3, 3, 3);

To ensure the database remains accurate and up to date I will ensure the usage of data validation methods to restrict the number of data types and characters for each “box” of a row. I will ensure that records are not added, unless new, and the use of modify/delete to modify records or delete them.

Due to the business being small the data will be stored locally and backups can be done automatically on a network drive, which can have storage added to it manually whenever needed.

Day 4: Task 2: SQL Practical

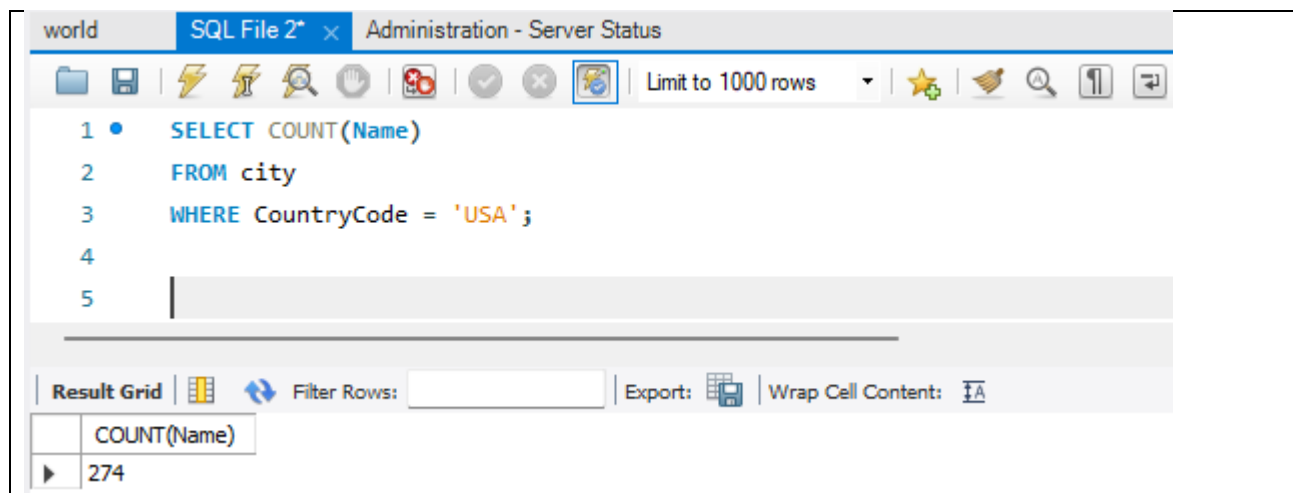
In your groups, work together to answer the below questions. It may be of benefit if one of you shares your screen with the group and as a team answer / take screen shots from there.

Setting up the database:

1. Download world_db(1) [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.



The screenshot shows a SQL query editor window with the following content:

```
1 • SELECT COUNT(Name)
2   FROM city
3   WHERE CountryCode = 'USA';
4
5
```

Below the query editor, the result grid is displayed:

	COUNT(Name)
▶	274

The interface includes a toolbar with various icons and a status bar at the bottom with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

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.

```

5 • SELECT * FROM country
6 ORDER BY LifeExpectancy DESC;

```

	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld	LocalName
▶	AND	Andorra	Europe	Southern Europe	468.00	1278	78000	83.5	1630.00	NULL	Andorra
	MAC	Macao	Asia	Eastern Asia	18.00	NULL	473000	81.6	5749.00	5940.00	Macau/Aomen
	SMR	San Marino	Europe	Southern Europe	61.00	885	27000	81.1	510.00	NULL	San Marino
	JPN	Japan	Asia	Eastern Asia	377829.00	-660	126714000	80.7	3787042.00	4192638.00	Nihon/Nippon
	SGP	Singapore	Asia	Southeast Asia	618.00	1965	3567000	80.1	86503.00	96318.00	Singapore/Singapura
	AUS	Australia	Oceania	Australia and New Zealand	7741220.00	1901	18886000	79.8	351182.00	392911.00	Australia
	CHE	Switzerland	Europe	Western Europe	41284.00	1499	7160400	79.6	264478.00	256092.00	Schweiz/Suisse/Svizz
	SWE	Sweden	Europe	Nordic Countries	449964.00	836	8861400	79.6	226492.00	227757.00	Sverige
	HKG	Hong Kong	Asia	Eastern Asia	1075.00	NULL	6782000	79.5	166448.00	173610.00	Xianggang/Hong Kong
	CAN	Canada	North America	North America	9970610.00	1867	31147000	79.4	598862.00	625626.00	Canada
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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.

```

7
8 • SELECT DISTINCT name
9 FROM city
10 WHERE name LIKE 'New%';

```

	name
▶	Newcastle
	Newcastle upon Tyne
	Newport
	New Bombay
	New Delhi
	New York
	New Orleans
	Newark
	Newport News
	New Haven
	New Bedford

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.

```

12 • SELECT * FROM country
13 ORDER BY Population DESC
14 LIMIT 0, 10;

```

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
CHN	China	Asia	Eastern Asia	9572900.00	-1523	1277558000
IND	India	Asia	Southern and Central Asia	3287263.00	1947	1013662000
USA	United States	North America	North America	9363520.00	1776	278357000
IDN	Indonesia	Asia	Southeast Asia	1904569.00	1945	212107000
BRA	Brazil	South America	South America	8547403.00	1822	170115000
PAK	Pakistan	Asia	Southern and Central Asia	796095.00	1947	156483000
RUS	Russian Federation	Europe	Eastern Europe	17075400.00	1991	146934000
BGD	Bangladesh	Asia	Southern and Central Asia	143998.00	1971	129155000
JPN	Japan	Asia	Eastern Asia	377829.00	-660	126714000
NGA	Nigeria	Africa	Western Africa	923768.00	1960	111506000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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.

```

16 • SELECT * FROM country
17 WHERE Population > 2000000
18 ORDER BY Population DESC;

```

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
CHN	China	Asia	Eastern Asia	9572900.00	-1523	1277558000
IND	India	Asia	Southern and Central Asia	3287263.00	1947	1013662000
USA	United States	North America	North America	9363520.00	1776	278357000
IDN	Indonesia	Asia	Southeast Asia	1904569.00	1945	212107000
BRA	Brazil	South America	South America	8547403.00	1822	170115000
PAK	Pakistan	Asia	Southern and Central Asia	796095.00	1947	156483000
RUS	Russian Federation	Europe	Eastern Europe	17075400.00	1991	146934000
BGD	Bangladesh	Asia	Southern and Central Asia	143998.00	1971	129155000
JPN	Japan	Asia	Eastern Asia	377829.00	-660	126714000
NGA	Nigeria	Africa	Western Africa	923768.00	1960	111506000
MEX	Mexico	North America	Central America	1958201.00	1810	98881000
DEU	Germany	Europe	Western Europe	357022.00	1955	82164700
VNM	Vietnam	Asia	Southeast Asia	331689.00	1945	79832000
PHL	Philippines	Asia	Southeast Asia	300000.00	1946	75967000
EGY	Egypt	Africa	Northern Africa	1001449.00	1922	68470000
IRN	Iran	Asia	Southern and Central Asia	1648195.00	1906	67702000
TUR	Turkey	Asia	Middle East	774815.00	1923	66591000
ETH	Ethiopia	Africa	Eastern Africa	1104300.00	-1000	62565000

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.

```
20 • SELECT *
21 FROM city
22 WHERE Name LIKE 'Be%';
```






	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
	453	Bento Gonçalves	BRA	Rio Grande d...	89254
	469	Belfast	GBR	North Ireland	287500
	724	Benoni	ZAF	Gauteng	365467
	949	Bekasi	IDN	West Java	644300
	980	Bengkulu	IDN	Bengkulu	146439

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.

```

24 • SELECT *
25 FROM city
26 WHERE Population >=500000 AND Population <=1000000;

```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
Export/Import:  					
	ID	Name	CountryCode	District	Population
▶	5	Amsterdam	NLD	Noord-Holland	731200
	6	Rotterdam	NLD	Zuid-Holland	593321
	36	Oran	DZA	Oran	609823
	64	Dubai	ARE	Dubai	669181
	72	Rosario	ARG	Santa Fé	907718
	73	Lomas de Zamora	ARG	Buenos Aires	622013
	74	Quilmes	ARG	Buenos Aires	559249
	75	Almirante Brown	ARG	Buenos Aires	538918
	76	La Plata	ARG	Buenos Aires	521936
	77	Mar del Plata	ARG	Buenos Aires	512880
	134	Adelaide	AUS	South Australia	978100
	152	Khulna	BGD	Khulna	663340
	186	Cotonou	BEN	Atlantique	536827
	193	Santa Cruz de la...	BOL	Santa Cruz	935361
	194	La Paz	BOL	La Paz	758141
	195	El Alto	BOL	La Paz	534466

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.

```

28 • SELECT Name AS 'City Name', CountryCode AS 'Country Code', District, Population
29 FROM city
30 ORDER BY Name ASC;

```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content: Fetch rows:				
City Name	Country Code	District	Population	
[San Cristóbal de] la Laguna	ESP	Canary Islands	127945	
's-Hertogenbosch	NLD	Noord-Brabant	129170	
A Coruña (La Coruña)	ESP	Galicia	243402	
Aachen	DEU	Nordrhein-Westfalen	243825	
Aalborg	DNK	Nordjylland	161161	
Aba	NGA	Imo & Abia	298900	
Abadan	IRN	Khuzestan	206073	
Abaetetuba	BRA	Pará	111258	
Abakan	RUS	Hakassia	169200	
Abbotsford	CAN	British Colombia	105403	
Abeokuta	NGA	Ogun	427400	
Aberdeen	GBR	Scotland	213070	
Abha	SAU	Asir	112300	
Abidjan	CIV	Abidjan	2500000	
Abiko	JPN	Chiba	126670	
Abilene	USA	Texas	115930	
Abohar	IND	Punjab	107163	
Abottabad	PAK	Nothwest Border Prov	106000	
Abu Dhabi	ARE	Abu Dhabi	398695	
Abuja	NGA	Federal Capital Dist	350100	
Acámbaro	MEX	Guanajuato	110487	
Acapulco de Juárez	MEX	Guerrero	721011	
Acarigua	VEN	Portuguesa	158954	
Accra	GHA	Greater Accra	1070000	
Achalpur	IND	Maharashtra	96216	
Acheng	CHN	Heilongjiang	197595	
Acuña	MEX	Coahuila de Zaragoza	110388	

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.



32	•	SELECT Name AS 'City Name', CountryCode AS 'Country Code', District, Population
33		FROM city
34		ORDER BY Population DESC
35		LIMIT 0, 20;

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




	City Name	Country Code	District	Population
▶	Mumbai (Bombay)	IND	Maharashtra	10500000
	Seoul	KOR	Seoul	9981619
	São Paulo	BRA	São Paulo	9968485
	Shanghai	CHN	Shanghai	9696300
	Jakarta	IDN	Jakarta Raya	9604900
	Karachi	PAK	Sindh	9269265
	Istanbul	TUR	Istanbul	8787958
	Ciudad de México	MEX	Distrito Federal	8591309
	Moscow	RUS	Moscow (City)	8389200
	New York	USA	New York	8008278
	Tokyo	JPN	Tokyo-to	7980230
	Peking	CHN	Peking	7472000
	London	GBR	England	7285000
	Delhi	IND	Delhi	7206704
	Cairo	EGY	Kairo	6789479
	Teheran	IRN	Teheran	6758845
	Lima	PER	Lima	6464693
	Chongqing	CHN	Chongqing	6351600
	Bangkok	THA	Bangkok	6320174
	Santafé de Bogotá	COL	Santafé de Bo...	6260862

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.

```

37 • SELECT COUNT(distinct Name), Name
38 FROM city
39 GROUP BY Name
40 ORDER BY Name ASC;

```

Result Grid |   Filter Rows: | Export: 

COUNT(distinct Name)	Name
1	[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	Abiko
1	Abilene
1	Abohar
1	Abottabad

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.




```

42 • SELECT Name AS 'City Name', CountryCode AS 'Country Code', District, Population
43 FROM city
44 ORDER BY Population ASC;

```

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

	City Name	Country Code	District	Population
▶	Adamstown	PCN	–	42
	West Island	CCK	West Island	167
	Fakaofo	TKL	Fakaofo	300
	Città del Vaticano	VAT	–	455
	Bantam	CCK	Home Island	503
	Yaren	NRU	–	559
	The Valley	AIA	–	595
	Alofi	NIU	–	682
	Flying Fish Cove	CXR	–	700
	Kingston	NFK	–	800
	South Hill	AIA	–	961
	Mata-Utu	WLF	Wallis	1137
	Agaña	GUM	–	1139

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.

```

46 • SELECT *
47 FROM country
48 ORDER BY Population DESC
49 LIMIT 0, 1;

```

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

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
CHN	China	Asia	Eastern Asia	9572900.00	-1523	1277558000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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.

```

50
51 • SELECT co.Code, co.Continent, co.Name AS Country, co.LocalName, ci.Name AS Capital
52 FROM country AS co, city AS ci
53 WHERE co.Capital=ci.ID AND co.Name = 'Spain';

```

Result Grid		Filter Rows:		Export:		Wrap Cell Content:	
	Code	Continent	Country	LocalName	Capital		
▶	ESP	Europe	Spain	España	Madrid		

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.

```

55 • SELECT co.Code, co.Continent, co.Name AS Country, co.LocalName, ci.Name AS City
56 FROM country AS co, city AS ci
57 WHERE ci.CountryCode=co.Code AND co.Continent='Europe';

```

Result Grid		Filter Rows:		Export:		Wrap Cell Content:	
	Code	Continent	Country	LocalName	City		
▶	ALB	Europe	Albania	Shqipëria	Tirana		
	AND	Europe	Andorra	Andorra	Andorra la Vella		
	AUT	Europe	Austria	Österreich	Wien		
	AUT	Europe	Austria	Österreich	Graz		
	AUT	Europe	Austria	Österreich	Linz		
	AUT	Europe	Austria	Österreich	Salzburg		
	AUT	Europe	Austria	Österreich	Innsbruck		
	AUT	Europe	Austria	Österreich	Klagenfurt		
	BEL	Europe	Belgium	België/Belgique	Antwerpen		
	BEL	Europe	Belgium	België/Belgique	Gent		
	BEL	Europe	Belgium	België/Belgique	Charleroi		
	BEL	Europe	Belgium	België/Belgique	Liège		
	BEL	Europe	Belgium	België/Belgique	Bruxelles [Brussel]		
	BEL	Europe	Belgium	België/Belgique	Brugge		
	BEL	Europe	Belgium	België/Belgique	Schaerbeek		
	BEL	Europe	Belgium	België/Belgique	Namur		
	BEL	Europe	Belgium	België/Belgique	Mons		
	BGR	Europe	Bulgaria	Balgarija	Sofija		
	BGR	Europe	Bulgaria	Balgarija	Ploudiv		

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.



59

•

SELECT AVG(Population), Name AS 'Country Name'

60

FROM country

61

GROUP BY(name);

Result Grid

Filter Rows:

Export:

Wrap C

	AVG(Population)	Country Name
▶	103000.0000	Aruba
	22720000.0000	Afghanistan
	12878000.0000	Angola
	8000.0000	Anguilla
	3401200.0000	Albania
	78000.0000	Andorra
	217000.0000	Netherlands Antilles
	2441000.0000	United Arab Emirates
	37032000.0000	Argentina
	3520000.0000	Armenia
	68000.0000	American Samoa
	0.0000	Antarctica
	0.0000	French Southern ter...
	68000.0000	Antigua and Barbuda
	18886000.0000	Australia
	8091800.0000	Austria
	7734000.0000	Azerbaijan
	6695000.0000	Burundi
	10730000.0000	Belgium

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.

```

63 • SELECT co.Code, co.Continent, co.Name AS Country, ci.Name AS Capital, ci.Population
64 FROM country AS co, city AS ci
65 WHERE co.Capital=ci.ID
66 ORDER BY ci.Population ASC
67 /*ORDER BY ci.Population DESC*/;

```

Result Grid

 Filter Rows:

 Export:

 Wrap Cell Content: ☐




	Code	Continent	Country	Capital	Population
▶	PCN	Oceania	Pitcairn	Adamstown	42
	CCK	Oceania	Cocos (Keeling) Islands	West Island	167
	TKL	Oceania	Tokelau	Fakaofu	300
	VAT	Europe	Holy See (Vatican City State)	Città del Vaticano	455
	NRU	Oceania	Nauru	Yaren	559
	AIA	North America	Anguilla	The Valley	595
	NIU	Oceania	Niue	Alofi	682
	CXR	Oceania	Christmas Island	Flying Fish Cove	700
	NFK	Oceania	Norfolk Island	Kingston	800
	WLF	Oceania	Wallis and Futuna	Mata-Utu	1137
	GUM	Oceania	Guam	Agaña	1139
	BMU	North America	Bermuda	Hamilton	1200
	MCO	Europe	Monaco	Monaco-Ville	1234
	SJM	Europe	Svalbard and Jan Mayen	Longyearbyen	1438
	SHN	Africa	Saint Helena	Jamestown	1500
	FLK	South America	Falkland Islands	Stanley	1636
	MSR	North America	Montserrat	Plymouth	2000
	KIR	Oceania	Kiribati	Bairiki	2226
	SMD	Europe	San Marino	San Marino	2794

Result 53 x

```

63 • SELECT co.Code, co.Continent, co.Name AS Country, ci.Name AS Capital, ci.Population
64 FROM country AS co, city AS ci
65 WHERE co.Capital=ci.ID
66 /*ORDER BY ci.Population ASC*/
67 ORDER BY ci.Population DESC;

```

Result Grid	 Filter Rows:	Export: 	Wrap Cell Content: 		
	Code	Continent	Country	Capital	Population
▶	KOR	Asia	South Korea	Seoul	9981619
	IDN	Asia	Indonesia	Jakarta	9604900
	MEX	North America	Mexico	Ciudad de México	8591309
	RUS	Europe	Russian Federation	Moscow	8389200
	JPN	Asia	Japan	Tokyo	7980230
	CHN	Asia	China	Peking	7472000
	GBR	Europe	United Kingdom	London	7285000
	EGY	Africa	Egypt	Cairo	6789479
	IRN	Asia	Iran	Teheran	6758845
	PER	South America	Peru	Lima	6464693
	THA	Asia	Thailand	Bangkok	6320174
	COL	South America	Colombia	Santafé de Bogotá	6260862
	COD	Africa	Congo, The Demo...	Kinshasa	5064000
	CHL	South America	Chile	Santiago de Chile	4703954
	IRQ	Asia	Iraq	Baghdad	4336000
	SGP	Asia	Singapore	Singapore	4017733
	BGD	Asia	Bangladesh	Dhaka	3612850
	DEU	Europe	Germany	Berlin	3386667
	MMR	Asia	Myanmar	Yangon (Naypyi Taw)	3361700

Result 54 x



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.

```
85  /*17*/
86  •  SELECT (SUM(Population)/SUM(SurfaceArea)) AS 'Population Density', Name AS 'Country Name'
87  FROM country
88  GROUP BY(name)
89  ORDER BY (SUM(Population)/SUM(SurfaceArea)) ASC;
90
```


Population Density	Country Name
0.0000	Antarctica
0.0000	French Southern territories
0.0000	Bouvet Island
0.0000	Heard Island and McDonald Islands
0.0000	British Indian Ocean Territory
0.0000	South Georgia and the South Sandwich Islands
0.0000	United States Minor Outlying Islands
0.0259	Greenland
0.0513	Svalbard and Jan Mayen
0.1643	Falkland Islands
1.0204	Pitcairn
1.1015	Western Sahara
1.6993	Mongolia
2.0111	French Guiana
2.0939	Namibia
2.4397	Australia
2.5541	Suriname
2.6036	Mauritania

18. **Countries 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.

```

91  /*18*/
92  • SELECT Name AS 'Country Name', (GNP/Population) AS 'GDP per Capita'
93  FROM country
94  WHERE (GNP/Population)>(
95  SELECT AVG(GNP/Population)
96  FROM country)
97  ORDER BY (GNP/Population) DESC;
98

```

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

	Country Name	GDP per Capita
▶	Luxembourg	0.037459
	Switzerland	0.036936
	Bermuda	0.035815
	Brunei	0.035686
	Liechtenstein	0.034644
	Cayman Islands	0.033237
	Denmark	0.032664
	Norway	0.032577
	United States	0.030575
	Japan	0.029887
	Iceland	0.029588
	Virgin Islands, ...	0.029143
	Austria	0.026182
	Germany	0.025965
	Sweden	0.025559
	Hong Kong	0.024543
	Belgium	0.024388
	Singapore	0.024251
	France	0.024048
	Finland	0.023575

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.

```

96  /**/
97  • SELECT Name AS 'City Name', CountryCode AS 'Country Code', District, Population
98  FROM city
99  ORDER BY Population DESC
100  LIMIT 31, 40;

```

Result Grid				
Filter Rows:		Export:		
Wrap Cell Content:				
	City Name	Country Code	District	Population
▶	Kanton [Guangzhou]	CHN	Guangdong	4256300
	Singapore	SGP	–	4017733
	Ho Chi Minh City	VNM	Ho Chi Minh City	3980000
	Chennai (Madras)	IND	Tamil Nadu	3841396
	Pusan	KOR	Pusan	3804522
	Los Angeles	USA	California	3694820
	Dhaka	BGD	Dhaka	3612850
	Berlin	DEU	Berlin	3386667
	Rangoon (Yangon)	MMR	Rangoon [Yangon]	3361700
	Chengdu	CHN	Sichuan	3361500
	Jokohama [Yokohama]	JPN	Kanagawa	3339594
	Alexandria	EGY	Aleksandria	3328196
	Riyadh	SAU	Riyadh	3324000
	Sydney	AUS	New South Wales	3276207
	Ankara	TUR	Ankara	3038159
	Buenos Aires	ARG	Distrito Federal	2982146
	Hyderabad	IND	Andhra Pradesh	2964638
	Casablanca	MAR	Casablanca	2940623
	Chicago	USA	Illinois	2896016
	Madrid	ESP	Madrid	2879052

city 65 x

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.

