By Kareem Shaaban

SQL

FIRST WEEK ASSIGNMENT WITH TDI

SQL

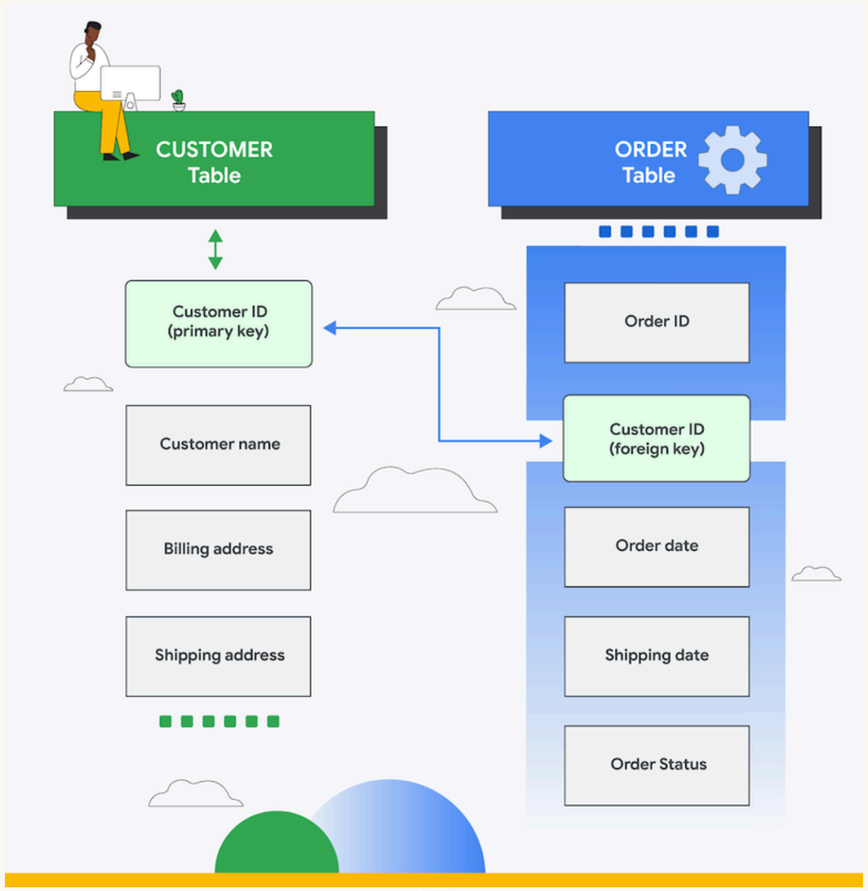
SQL (Structured Query Language) is incredibly beneficial in many scenarios where data needs to be retrieved, manipulated, or analyzed from a relational database.

SQL

SQL contributes to data integrity by enforcing rules and constraints through primary/foreign keys, constraints like NOT NULL and UNIQUE, data types, transactions, and referential integrity. These features ensure the data in the database remains accurate, consistent, and reliable over time.

DBMS

A Database Management System (DBMS) is software that allows users to create, manage, and manipulate databases efficiently. utilizing SQL as the language to interact with the data.



One to many relationship

RDB

A relational database organizes data into tables, which consist of rows and columns. This structure allows for efficient data management and retrieval. Each table typically represents a different entity, and relationships between tables are established using keys (primary and foreign keys). This model supports SQL (Structured Query Language) for querying and maintaining the database.

Select

1. Write an SQL query to select all columns from the table named locations

2. Retrieve only the country and population columns from the table.

Query (1)

```
select * from locations;
```

Output (1)

	location_id	region	country	population	density
1	101	Northland	New Zealand	201500	16.1100006103516
2	102	Auckland	New Zealand	1695200	343.089996337891
3	103	Waikato	New Zealand	513800	21.5
4	104	Bay of Plenty	New Zealand	347700	28.7999992370605
5	105	Gisborne	New Zealand	52100	6.21000003814697
6	106	Hawke's Bay	New Zealand	182700	12.9200000762939
7	107	Taranaki	New Zealand	127300	17.5499992370605
8	108	Manawatū-...	New Zealand	258200	11.6199998855591
9	109	Wellington	New Zealand	543500	67.5199966430664
10	110	Tasman	New Zealand	58700	6.09999990463257
11	111	Nelson	New Zealand	54500	129.149993896484
12	112	Marlborough	New Zealand	51900	4.94000005722046
13	113	West Coast	New Zealand	32700	1.4099999666214
14	114	Canterbury	New Zealand	655000	14.7200002670288
15	115	Otago	New Zealand	246000	7.8899998664856
16	116	Southland	New Zealand	102400	3.27999997138977

MS SQL server view

Select

1. Write an SQL query to select all columns from the table named locations

2. Retrieve only the country and population columns from the table.

Query (2)

```
select country , population from locations;
```

Output (2)

	country	population
1	New Zealand	201500
2	New Zealand	1695200
3	New Zealand	513800
4	New Zealand	347700
5	New Zealand	52100
6	New Zealand	182700
7	New Zealand	127300
8	New Zealand	258200
9	New Zealand	543500
10	New Zealand	58700
11	New Zealand	54500
12	New Zealand	51900
13	New Zealand	32700
14	New Zealand	655000
15	New Zealand	246000
16	New Zealand	102400

MS SQL server view

From

1 In the context of your table, how would you structure a query to select data from the locations table?

By using Select statement

Query

```
select * from locations;
```

Output

	location_id	region	country	population	density
1	101	Northland	New Zealand	201500	16.1100006103516
2	102	Auckland	New Zealand	1695200	343.089996337891
3	103	Waikato	New Zealand	513800	21.5
4	104	Bay of Plenty	New Zealand	347700	28.7999992370605
5	105	Gisborne	New Zealand	52100	6.21000003814697
6	106	Hawke's Bay	New Zealand	182700	12.9200000762939
7	107	Taranaki	New Zealand	127300	17.5499992370605
8	108	Manawatū-...	New Zealand	258200	11.6199998855591
9	109	Wellington	New Zealand	543500	67.5199966430664
10	110	Tasman	New Zealand	58700	6.09999990463257
11	111	Nelson	New Zealand	54500	129.149993896484
12	112	Marlborough	New Zealand	51900	4.94000005722046
13	113	West Coast	New Zealand	32700	1.4099999666214
14	114	Canterbury	New Zealand	655000	14.7200002670288
15	115	Otago	New Zealand	246000	7.8899998664856
16	116	Southland	New Zealand	102400	3.27999997138977

MS SQL server view

Where

1. Write an SQL query to select regions where the population is greater than 500,000
2. Write an SQL query to select regions with a population density less than 50
3. Retrieve data for countries in the Canterbury region

Query (1)

```
select region from locations where population > 500000;
```

Output (1)

	region
1	Auckland
2	Waikato
3	Wellington
4	Canterbury

MS SQL server view

Where

1. Write an SQL query to select regions where the population is greater than 500,000
2. Write an SQL query to select regions with a population density less than 50
3. Retrieve data for countries in the Canterbury region

Query (2)

```
select region from locations where density < 50;
```

Output (2)

	region
1	Northland
2	Waikato
3	Bay of Plenty
4	Gisborne
5	Hawke's Bay
6	Taranaki
7	Manawatū-Whanganui
8	Tasman
9	Marlborough
10	West Coast
11	Canterbury
12	Otago
13	Southland

MS SQL server view

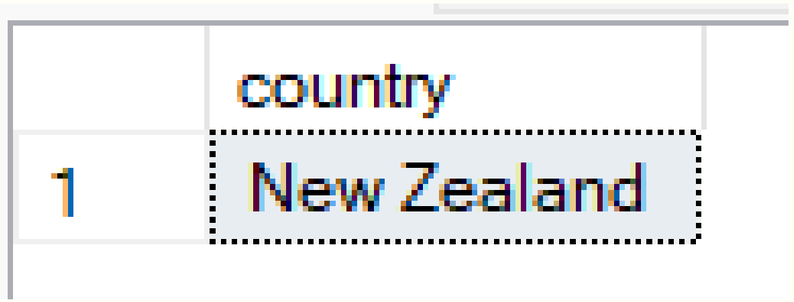
Where

1. Write an SQL query to select regions where the population is greater than 500,000
2. Write an SQL query to select regions with a population density less than 50
3. Retrieve data for countries in the Canterbury region

Query (3)

```
select country from locations where region = 'Canterbury';
```

Output (3)



	country
1	New Zealand

MS SQL server view

Order By

1. How would you sort the results to display Regions in descending order based on population?
2. Sort the records alphabetically by region
3. Retrieve regions ordered by their population density in descending order but only include countries with a population greater than 500,000

Query (1)

```
select region , population from locations order by population desc ;
```

Output (1)

	region	population
1	Auckland	1695200
2	Canterbury	655000
3	Wellington	543500
4	Waikato	513800
5	Bay of Plenty	347700
6	Manawatū-Whanganui	258200
7	Otago	246000
8	Northland	201500
9	Hawke's Bay	182700
10	Taranaki	127300
11	Southland	102400
12	Tasman	58700
13	Nelson	54500
14	Gisborne	52100
15	Marlborough	51900
16	West Coast	32700

MS SQL server view

Order By

1. How would you sort the results to display Regions in descending order based on population?
2. Sort the records alphabetically by region
3. Retrieve regions ordered by their population density in descending order but only include countries with a population greater than 500,000

Query (2)

```
select * from locations order by region asc ;
```

Output (2)

	location_id	region	country	population	density
1	102	Auckland	New Zealand	1695200	343.089996337891
2	104	Bay of Plenty	New Zealand	347700	28.7999992370605
3	114	Canterbury	New Zealand	655000	14.7200002670288
4	105	Gisborne	New Zealand	52100	6.21000003814697
5	106	Hawke's Bay	New Zealand	182700	12.9200000762939
6	108	Manawatū-Whanganui	New Zealand	258200	11.6199998855591
7	112	Marlborough	New Zealand	51900	4.94000005722046
8	111	Nelson	New Zealand	54500	129.149993896484
9	101	Northland	New Zealand	201500	16.1100006103516
10	115	Otago	New Zealand	246000	7.8899998664856
11	116	Southland	New Zealand	102400	3.27999997138977
12	107	Taranaki	New Zealand	127300	17.5499992370605
13	110	Tasman	New Zealand	58700	6.09999990463257
14	103	Waikato	New Zealand	513800	21.5
15	109	Wellington	New Zealand	543500	67.5199966430664
16	113	West Coast	New Zealand	32700	1.4099999666214

MS SQL server view

Order By

1. How would you sort the results to display Regions in descending order based on population?
2. Sort the records alphabetically by region
3. Retrieve regions ordered by their population density in descending order but only include countries with a population greater than 500,000

Query (3)

```
select region , population , density from locations where population > 500000 order by density desc;
```

Output (3)

	region	population	density
1	Auckland	1695200	343.089996337891
2	Wellington	543500	67.5199966430664
3	Waikato	513800	21.5
4	Canterbury	655000	14.7200002670288

MS SQL server view

Group By

1. Retrieve the country with total population.

Query

```
select country , sum (population) as population from locations group by country ;
```

Output

	country	population
1	New Zealand	5123200

MS SQL server view

THANK YOU!

Presented By Kareem Shaaban