UP940148 DBPRIN Submission 4

Every query I made has two versions:

- First I have the version that I would recommend be run when querying the database manually.
- Second I have the version that I would recommend be run when querying the database through a script (JavaScript was in mind).

After each query is a screenshot showing the data that was retrieved

Query 1. A query to get all the useful information regarding a customer's order

Version 1

```
SELECT
Orders.orderId,
CONCAT(Customer.forename, ' ', Customer.surname) as "Customer Name",
Product.name as "Product",
Product.price as "Price",
Orders.delivery as "Delivery",
CONCAT(
  CASE WHEN Orders.delivery THEN
    Orders.addressLine1
  ELSE
    Location.addressLine1
  END,
  chr(10),
  CASE WHEN Orders.delivery THEN
    Orders.postcode
  ELSE
    Location.postcode
  END,
  chr(10),
  CASE WHEN Orders.delivery THEN
    Orders.country
  ELSE
    Location.country
  END
) as "Address"

FROM
Orders
inner join Product on Product.productId = Orders.productId
inner join Location on Location.locationId = Orders.locationId
inner join Customer on Customer.customerId = Orders.customerId

WHERE Orders.orderId = 12;
```

```
dbprincw1-# WHERE Orders.orderId = 12;
 orderid | Customer Name | Product  |  Price  | Delivery |    Address
---------+---------------+----------+---------+----------+----------------
      12 | Caz Sheppey   | Firebird | $415.46 | f        | 21742         +
         |               |          |         |          | DN36          +
         |               |          |         |          | United Kingdom
(1 row)
```

Version 2

```
SELECT
Orders.orderId,
Customer.forename,
Customer.surname,
Product.name,
Product.price,
```

```
Orders.delivery,
CASE WHEN Orders.delivery THEN
  Orders.addressLine1
ELSE
  Location.addressLine1
END,

CASE WHEN Orders.delivery THEN
  Orders.postcode
ELSE
  Location.postcode
END,

CASE WHEN Orders.delivery THEN
  Orders.country
ELSE
  Location.country
END

FROM
Orders
inner join Product on Product.productId = Orders.productId
inner join Location on Location.locationId = Orders.locationId
inner join Customer on Customer.customerId = Orders.customerId

WHERE Orders.orderId = 12;
```

```
dbprincw1-# WHERE Orders.orderId = 12;
 orderid | forename | surname |   name   |  price  | delivery | addressline1 | postcode |    country
---------+----------+---------+----------+---------+----------+--------------+----------+----------------
      12 | Caz      | Sheppey | Firebird | $415.46 | f        | 21742        | DN36     | United Kingdom
(1 row)
```

Query 2. Retrieves the revenue generated at each location between two set dates

Version 1

```
SELECT
Location.locationId as "Location",
SUM(Product.price) as "Revenue Generated"

FROM Orders
inner join Product on Product.productId = Orders.productId
inner join Location on Location.locationId = Orders.locationId

WHERE Orders.orderDate between TO_DATE('2020-12-01', 'YYYY-MM-DD') and TO_DATE('2021-01-01', 'YYYY-MM-DD')
/* Dates would be supplied by an external script */

GROUP BY Location.locationId
ORDER BY Location.locationId;
```

```
dbprincw1-# ORDER BY Location.locationId;
 Location | Revenue Generated
----------+------------------
        1 |         $1,926.28
        2 |         $1,612.89
        3 |           $870.77
        4 |         $1,385.89
        5 |           $829.94
        6 |           $410.34
        8 |         $2,009.87
        9 |         $2,352.32
       10 |           $298.86
(9 rows)
```

Version 2

```
SELECT
Location.locationId,
```

```sql
SUM(Product.price) as "revenue_generated"

FROM Orders
inner join Product on Product.productId = Orders.productId
inner join Location on Location.locationId = Orders.locationId

WHERE Orders.orderDate between TO_DATE('2020-12-01', 'YYYY-MM-DD') and TO_DATE('2021-01-01', 'YYYY-MM-DD')
/* Dates would be supplied by an external script */

GROUP BY Location.locationId
ORDER BY Location.locationId;
```

```
dbprincw1-# ORDER BY Location.locationId;
 locationid | revenue_generated
------------+------------------
          1 |         $1,926.28
          2 |         $1,612.89
          3 |           $870.77
          4 |         $1,385.89
          5 |           $829.94
          6 |           $410.34
          8 |         $2,009.87
          9 |         $2,352.32
         10 |           $298.86
(9 rows)
```

Query 3. Gets the availability of an item across all locations

Version 1

```sql
SELECT
Product.name as "Product",
Stock.stockCount as "Stock available",
Location.locationId as "Location",
CONCAT(
  Location.addressLine1,
  chr(10),
  Location.postcode,
  chr(10),
  Location.country
) as "Store Address"

FROM Stock
inner join Product on Product.productId = Stock.productId
inner join Location on Location.locationId = Stock.locationId

WHERE Product.productId = 40

ORDER BY Location.locationId;
```

```
dbprincw1-# ORDER BY Location.locationId;
 Product | Stock available | Location | Store Address
---------+-----------------+----------+---------------
 Sonata  |              11 |        1 | 7902          +
         |                 |          | BS14          +
         |                 |          | United Kingdom
 Sonata  |               1 |        2 | 6             +
         |                 |          | S1            +
         |                 |          | United Kingdom
 Sonata  |               7 |        3 | 97879         +
         |                 |          | RH5           +
         |                 |          | United Kingdom
 Sonata  |               8 |        4 | 234           +
         |                 |          | GU32          +
         |                 |          | United Kingdom
 Sonata  |              10 |        5 | 267           +
         |                 |          | CT16          +
         |                 |          | United Kingdom
 Sonata  |              10 |        6 | 34926         +
         |                 |          | BS41          +
         |                 |          | United Kingdom
 Sonata  |               5 |        7 | 4231          +
         |                 |          | LE14          +
         |                 |          | United Kingdom
 Sonata  |               7 |        8 | 98379         +
         |                 |          | RH5           +
         |                 |          | United Kingdom
 Sonata  |               7 |        9 | 4             +
         |                 |          | CT16          +
         |                 |          | United Kingdom
 Sonata  |               5 |       10 | 21742         +
         |                 |          | DN36          +
         |                 |          | United Kingdom
(10 rows)
```

Version 2

```
SELECT
Product.name,
Stock.stockCount,
Location.locationId,
Location.addressLine1,
Location.postcode,
Location.country

FROM Stock
inner join Product on Product.productId = Stock.productId
inner join Location on Location.locationId = Stock.locationId

WHERE Product.productId = 40

ORDER BY Location.locationId;
```

```
dbprincw1-# ORDER BY Location.locationId;
  name  | stockcount | locationid | addressline1 | postcode |    country
--------+------------+------------+--------------+----------+---------------
 Sonata |         11 |          1 | 7902         | BS14     | United Kingdom
 Sonata |          1 |          2 | 6            | S1       | United Kingdom
 Sonata |          7 |          3 | 97879        | RH5      | United Kingdom
 Sonata |          8 |          4 | 234          | GU32     | United Kingdom
 Sonata |         10 |          5 | 267          | CT16     | United Kingdom
 Sonata |         10 |          6 | 34926        | BS41     | United Kingdom
 Sonata |          5 |          7 | 4231         | LE14     | United Kingdom
 Sonata |          7 |          8 | 98379        | RH5      | United Kingdom
 Sonata |          7 |          9 | 4            | CT16     | United Kingdom
 Sonata |          5 |         10 | 21742        | DN36     | United Kingdom
(10 rows)
```

Query 4. Gets all the staff that work at a given location. Shows the upcoming leavers at the top of the list

Version 1

```sql
SELECT
Staff.staffId as "Employee ID",
CONCAT(Staff.forename, ' ', Staff.surname) as "Name",
PartTime.endDate as "End date"

FROM Staff
inner join Location on Location.locationId = Staff.locationId
full join PartTime on PartTime.staffId = Staff.staffId

WHERE Location.locationId = 5
ORDER BY "End date", "Employee ID";
/* Orders them so that part time staff show up first, and the first one leaving can be seen clearly */
```

```
dbprincw1-# ORDER BY "End date", "Employee ID";
 Employee ID |       Name        |  End date
-------------+-------------------+------------
         101 | Catlee Iacomini   | 2021-03-02
         125 | Ikey Bonnefin     | 2021-03-05
         110 | Chloe Maceur      | 2021-04-05
         117 | Ursola Sonier     | 2021-04-08
         132 | Farly Thistleton  |
         133 | Bonnee Ainley     |
         149 | Dulciana Tomson   |
         156 | Prentiss Wenman   |
         158 | Lambert Dowles    |
         160 | Manolo Cokayne    |
         167 | Philippine Peer   |
         180 | Hazel Fernan      |
         182 | Phil Argile       |
         193 | Jessee Ballefant  |
         198 | Brooke Wildblood  |
(15 rows)
```

Version 2

```sql
SELECT
Staff.staffId,
Staff.forename,
Staff.surname,
PartTime.endDate

FROM Staff
inner join Location on Location.locationId = Staff.locationId
full join PartTime on PartTime.staffId = Staff.staffId

WHERE Location.locationId = 5
ORDER BY PartTime.endDate, Staff.staffId;
/* Orders them so that part time staff show up first, and the first one leaving can be seen clearly */
```

```
dbprincw1-# ORDER BY PartTime.endDate, Staff.staffId;
 staffid |  forename   |  surname    |  enddate
---------+-------------+-------------+------------
     101 | Catlee      | Iacomini    | 2021-03-02
     125 | Ikey        | Bonnefin    | 2021-03-05
     110 | Chloe       | Maceur      | 2021-04-05
     117 | Ursola      | Sonier      | 2021-04-08
     132 | Farly       | Thistleton  |
     133 | Bonnee      | Ainley      |
     149 | Dulciana    | Tomson      |
     156 | Prentiss    | Wenman      |
     158 | Lambert     | Dowles      |
     160 | Manolo      | Cokayne     |
     167 | Philippine  | Peer        |
     180 | Hazel       | Fernan      |
     182 | Phil        | Argile      |
     193 | Jessee      | Ballefant   |
     198 | Brooke      | Wildblood   |
(15 rows)
```

Query 5. Retrieves a sales report for a store between 2 dates

Version 1

```
SELECT
DISTINCT
Product.productId as "Product ID",
Product.name as "Item",
(
  SELECT COUNT(*)
  FROM Orders
  WHERE Orders.productId = Product.productId
  AND Orders.orderDate between TO_DATE('2020-12-01', 'YYYY-MM-DD') and TO_DATE('2021-01-01', 'YYYY-MM-DD')
) as "Sales",

(
  SELECT COUNT(*)
  FROM Orders
  WHERE Orders.productId = Product.productId
  AND Orders.orderDate between TO_DATE('2020-12-01', 'YYYY-MM-DD') and TO_DATE('2021-01-01', 'YYYY-MM-DD')
)*Product.price as "Revenue Gained"

FROM Orders
inner join Product on Product.productId = Orders.productId
inner join Location on Location.locationId = Orders.locationId

WHERE Location.locationId = 8
AND Orders.orderDate between TO_DATE('2020-12-01', 'YYYY-MM-DD') and TO_DATE('2021-01-01', 'YYYY-MM-DD')

GROUP BY Product.productId
ORDER BY "Revenue Gained" DESC;
```

```
dbprincw1-# ORDER BY "Revenue Gained" DESC;
 Product ID |    Item    | Sales | Revenue Gained
------------+------------+-------+----------------
         20 | LR2        |     2 |      $1,190.24
         13 | Falcon     |     2 |      $1,092.82
         48 | Millenia   |     1 |        $661.24
         23 | Boxster    |     2 |        $414.20
(4 rows)
```

Version 2

```
SELECT
DISTINCT
Product.productId,
Product.name,
(
  SELECT COUNT(*)
  FROM Orders
  WHERE Orders.productId = Product.productId
  AND Orders.orderDate between TO_DATE('2020-12-01', 'YYYY-MM-DD') and TO_DATE('2021-01-01', 'YYYY-MM-DD')
) as "total_sales",

(
  SELECT COUNT(*)
  FROM Orders
  WHERE Orders.productId = Product.productId
  AND Orders.orderDate between TO_DATE('2020-12-01', 'YYYY-MM-DD') and TO_DATE('2021-01-01', 'YYYY-MM-DD')
)*Product.price as "revenue_gained"

FROM Orders
inner join Product on Product.productId = Orders.productId
inner join Location on Location.locationId = Orders.locationId

WHERE Location.locationId = 8
AND Orders.orderDate between TO_DATE('2020-12-01', 'YYYY-MM-DD') and TO_DATE('2021-01-01', 'YYYY-MM-DD')
```

```sql
GROUP BY Product.productId
ORDER BY "revenue_gained" DESC;
```

```
dbprincw1-# ORDER BY "revenue_gained" DESC;
 productid |   name   | total_sales | revenue_gained
-----------+----------+-------------+----------------
        20 | LR2      |           2 |      $1,190.24
        13 | Falcon   |           2 |      $1,092.82
        48 | Millenia |           1 |        $661.24
        23 | Boxster  |           2 |        $414.20
(4 rows)
```