

CZ2007 Database Project

The repository contains scripts related to the final lab for CZ2007 through Microsoft SQL.

Prerequisite

[Microsoft SQL](#)

Importing Database

Through any Microsoft SQL server

1. Right click Databases
2. Select 'Import Data-tier Application'
3. Under 'Import from local disk', select [dsaig6.bacpac file](#)
4. Enter database name and wait for import

(Optional) Accessing SWLAB2 SQL server (VScode)

For quick access to database through VSCode.

1. Ensure the following tools are available
 1. NTUwireless VPN access
 2. [Visual Studio Code](#)
 3. [SQL Server \(mssql\) extension](#)
2. Add following snippet to VSCode **settings.json** for quick connection ([for more info](#))

```
"mssql.connections": [  
  {  
    "server": "155.69.100.36",  
    "database": "dsaig6",  
    "authenticationType": "SqlLogin",  
    "user": "dsaig6",  
    "password": "password",  
    "savePassword": true,  
    "profileName": "CZ2007_Database"  
  }  
]
```

Database Scripts

The following .sql scripts used in the making of the overall database.

- [Link to creation scripts](#)
- [Link to example scripts](#)
- [Link to selection scripts](#)

- [Link to exercise queries scripts](#)
- [Link to deletion scripts](#)

Required Exercise Queries

As required for the submission, the following scripts are in accordance with Appendix B.

Appendix B Query 1

Given a customer by an email address, returns the product ids that have been ordered and paid by this customer but not yet shipped.

```
SELECT p.ProductID
FROM Product p
    INNER JOIN OrderItem oi ON p.ProductID = oi.ProductID
    INNER JOIN Orders o ON o.OrderID = oi.OrderID
    INNER JOIN Customer c ON c.CustomerID = o.CustomerID
    INNER JOIN Invoice i ON i.InvoiceNumber = o.InvoiceNumber
WHERE oi.ItemStatus <> 'shipped' AND i.InvoiceStatus = 'paid'
AND o.CustomerID = (SELECT CustomerID
FROM Customer
WHERE Email = 'benedict85@hotmail.com')
;
```

Appendix B Query 2

Find the 3 bestselling product type ids in terms of product quantity sold. The products of concerned have to be ordered and paid. Whether they have been shipped is irrelevant.

```
SELECT TOP 3
    (oi.ProductID) , SUM(oi.Quantity) AS TotalQuantity
FROM ProductType pt
    INNER JOIN Product p ON p.ProductTypeID = pt.ProductTypeID
    INNER JOIN OrderItem oi ON oi.ProductID = p.ProductID
    INNER JOIN Orders o ON o.OrderID = oi.OrderID
    INNER JOIN Invoice i ON i.InvoiceNumber = o.InvoiceNumber
WHERE i.InvoiceStatus = 'paid'
GROUP BY oi.ProductID
ORDER BY TotalQuantity DESC
;
```

Appendix B Query 3

Return the descriptions of all the 2nd level product types. The product types with no parent will be regarded as 1st level product types and their direct child product types will be regarded as 2nd level.

```
SELECT pt.ProductTypeDesc
FROM ProductType pt
    INNER JOIN ProductType pt2 ON pt.ParentID = pt2.ProductTypeID
WHERE pt2.ParentID IS NULL
;
```

Appendix B Query 4

Find 2 product ids that are ordered together the most.

```
SELECT TOP 1
    (ProdMatches), ProductID1, ProductID2
FROM
    (SELECT COUNT(*) AS ProdMatches, ProductID1, ProductID2
    FROM
        (SELECT oi.ProductID AS ProductID1 , oi2.ProductID AS ProductID2
        FROM OrderItem oi
            INNER JOIN OrderItem oi2 ON oi2.OrderID = oi.OrderID
            WHERE oi2.ProductID <> oi.ProductID AND oi2.OrderID = oi.OrderID) AS d
    GROUP BY ProductID1, ProductID2) AS d2
;
```

Appendix B Query 5

Get 3 random customers and return their email addresses.

```
SELECT TOP 3
    Email
FROM Customer
ORDER BY NEWID()
;
```

Personalised Query 1

Find customers that have one paid for at least an item from each restricted shop.

```
SELECT c.FullName
FROM Customer c
    JOIN Orders o ON o.CustomerID = c.CustomerID
    JOIN CreditCard cc ON cc.CustomerID = c.CustomerID
    JOIN OrderItem oi ON oi.OrderID = o.OrderID
    JOIN Invoice i ON i.InvoiceNumber = o.InvoiceNumber
    JOIN Product p ON p.ProductID = oi.ProductID
    JOIN RestrictedShop rs ON rs.ProductTypeID = p.ProductTypeID
WHERE i.InvoiceStatus = 'paid'
GROUP BY c.FullName
```

```
HAVING count(distinct rs.ShopID) = (SELECT count(*)  
FROM RestrictedShop)  
;
```

Personalised Query 2

Obtain gross sales for the shipped products of shops in the month of February.

```
SELECT p.ShopID, SUM(oi.UnitPrice * oi.Quantity) AS GrossSale  
FROM OrderItem AS oi  
    JOIN Product AS p ON oi.ProductID = p.ProductID  
    JOIN Orders AS o ON o.OrderID = oi.OrderID  
WHERE month(OrderDate)='2' AND o.OrderStatus = 'shipped'  
GROUP BY p.ShopID  
ORDER BY GrossSale DESC  
;
```

Authors

- [U1822304C Qwek Zhi Hui](#)



◦

- [U1822199A Mok Wei Min](#)




◦

- [U1822815K Ng Jin Han Benedict](#)



◦

- [U1820136A Png Jun Sheng](#)



◦

Contributions

Lab 1

| Name | Individual Contribution | Percentage of Contribution |
|-------------|-----------------------------------|-----------------------------------|
| Zhi Hui | Discussion, Conversion to digital | 25 |
| Wei min | Discussion | 25 |
| Benedict | Discussion | 25 |
| Jun Sheng | Discussion | 25 |

Lab 3

| Name | Individual Contribution | Percentage of Contribution |
|-------------|-----------------------------------|-----------------------------------|
| Zhi Hui | Discussion | 25 |
| Wei min | Initial edits, Discussion | 25 |
| Benedict | Discussion, Conversion to digital | 25 |
| Jun Sheng | Discussion, Conversion to digital | 25 |

Lab 5

| Name | Individual Contribution | Percentage of Contribution |
|-------------|---|-----------------------------------|
| Zhi Hui | DB Setup, Intial edits, Discussion | 25 |
| Wei min | Intial edits, Query formation, Discussion | 25 |
| Benedict | Query Formation, Discussion | 25 |
| Jun Sheng | Discussion | 25 |