

Database SQL Recap

Select Query - basic

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the connection is to 'cssql.jh1033 (UNIVERSITY\jh1033 (66))'. The Object Explorer on the left shows the database structure, including tables like 'dbo.eCust', 'dbo.eTest', 'dbo.evehicle', 'dbo.Project', 'dbo.sausages', 'dbo.tBook', 'dbo.tcat', 'dbo.tcontact', and 'dbo.tcust'. The 'Columns' list for 'dbo.tcust' is expanded, showing fields like 'CustomerID', 'FirstName', 'LastName', 'EmailAddress', 'AddressLine1', 'City', 'PostalCode', 'CountryRegionName', and 'DeleteFlag'. The main query editor contains the following SQL code:

```
use jh1033
select * from tcust
```

Red text annotations are present: 'Move to your db' next to the 'use jh1033' line, and 'First query – all rows and all fields from table' below the 'select * from tcust' line. The 'Results' pane at the bottom displays a table with 10 rows and 10 columns. The first row is highlighted, showing '502' in the 'CustomerID' column. The status bar at the bottom indicates 'Query executed successfully.' and '402 rows'.

	CustomerID	FirstName	LastName	EmailAddress	AddressLine1	City	PostalCode	CountryRegionName	Delete
1	502	Hary	Robinson	hs@gmail.com	15 Acacia Ave	London	SE12RGU	UK	0
2	1	Georgina	Smith	eugene10@adventure-works.com	2243 W St.	Seaford	3198	Australia	0
3	2	Roger	Greene	ruben35@adventure-works.com	5844 Linden Land	Hobart	7001	Australia	0
4	3	Christy	Zhu	christy12@adventure-works.com	1825 Village Pl.	North Ryde	2113	Australia	0
5	4	Elizabeth	Johnson	elizabeth5@adventure-works.com	7553 Hamess Circle	Wollongong	2500	Australia	0
6	5	Julio	Ruiz	julio1@adventure-works.com	7305 Humphrey Drive	East Brisbane	4169	Australia	0
7	6	Janet	Alvarez	janet9@adventure-works.com	2612 Berry Dr	Matraville	2036	Australia	0
8	7	Marco	Mehta	marco14@adventure-works.com	942 Brook Street	Warrambool	3280	Australia	0
9	8	Rob	Verhoff	rob4@adventure-works.com	624 Peabody Road	Bendigo	3550	Australia	0
10	9	Shannon	Carlson	shannon38@adventure-works.com	3839 Northgate Road	Hervey Bay	4655	Australia	0

Select query – ordering results – must be final clause

SQLQuery1.sql - cssql.jh1033 (UNIVERSITY\jh1033 (66))* - Microsoft SQL Server Management Studio (Administrator)

File Edit View Query Project Debug Tools Window Help

Object Explorer

- Connect
- dbo.tcust3
- dbo.tDepartment
- dbo.torder
 - Columns
 - SalesOrderID (int, not null)
 - OrderDate (date, not null)
 - DueDate (date, not null)
 - ShipDate (date, null)
 - CustomerID (int, not null)
 - SalesPersonID (int, null)
 - Status (tinyint, not null)
 - value (money, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
- dbo.torderdetail
 - Columns
 - SalesOrderID (int, not null)
 - ProductID (int, not null)
 - OrderQty (smallint, null)
 - UnitPrice (money, null)
 - LineTotal (numeric(38,6),
 - Keys
 - Constraints
 - Triggers

SQLQuery1.sql - css...ERSITY\jh1033 (66))*

```
use jh1033
select * from tOrder
order by customerID
```

146 %

Results Messages

	SalesOrderID	OrderDate	DueDate	ShipDate	CustomerID	SalesPersonID	Status	value
1	49821	2003-04-25	2003-04-13	2003-04-08	1	275	5	10401789.46
2	43860	2001-08-01	2001-08-13	2001-08-08	1	280	5	1460373.93
3	44501	2001-11-01	2001-11-13	2001-11-08	1	280	5	2612886.74
4	44791	2001-12-01	2001-12-13	2001-12-08	1	277	5	1924866.66
5	45283	2002-02-01	2002-02-13	2002-02-08	1	280	5	3764313.78
6	46042	2002-05-01	2002-05-13	2002-05-08	1	280	5	3472299.06
7	46976	2002-08-01	2002-08-13	2002-08-08	2	283	5	1018407.74
8	47997	2002-11-01	2002-11-13	2002-11-08	2	283	5	546959.41
9	49054	2003-02-01	2003-02-13	2003-02-08	2	283	5	173940.78
10	69488	2004-05-01	2004-05-13	2004-05-08	2	283	5	90831.99
11	57044	2003-11-01	2003-11-13	2003-11-08	2	283	5	453784.84
12	63198	2004-02-01	2004-02-13	2004-02-08	2	283	5	405395.06
13	50216	2003-05-01	2003-05-13	2003-05-08	2	283	5	193551.66
14	51728	2003-08-01	2003-08-13	2003-08-08	2	283	5	390525.47
15	1	2010-12-01	2010-12-01	2010-12-01	2	NULL	5	55.00
16	50748	2003-06-01	2003-06-13	2003-06-08	3	275	5	5208749.30

Query executed successfully. | cssql (11.0 SP2) | UNIVERSITY\jh1033 (66) | jh1033 | 00:00:00 | 311 rows

Ready Ln 3 Col 20 Ch 20 INS

Select Query – some fields

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'jh1033', including tables and columns. The 'Columns' folder for the 'tCust' table is expanded, showing fields like 'CustomerID', 'FirstName', 'LastName', 'EmailAddress', 'AddressLine1', 'City', 'PostalCode', 'CountryRegionName', and 'DeleteFlag'. The main query window on the right shows the following SQL query:

```
use jh1033
select customerId, PostalCode from tCust
```

Below the query window, the 'Results' tab shows the output of the query, which is a table with two columns: 'customerId' and 'PostalCode'. The table contains 10 rows of data. The status bar at the bottom indicates that the query was executed successfully, returning 402 rows.

Choose only some fields
Separate field names with commas
Check field names by looking at the Object Explorer

	customerId	PostalCode
1	502	SE12RGU
2	1	3198
3	2	7001
4	3	2113
5	4	2500
6	5	4169
7	6	2036
8	7	3280
9	8	3550
10	9	4655

Query executed succ... | cssql (11.0 SP2) | UNIVERSITY\jh1033 (66) | jh1033 | 00:00:00 | 402 rows

Ready | Ln 2 | Col 41 | Ch 41 | INS

Select Query – some rows (records)– where clause using =

The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the connection is to 'UNIVERSITY\jh1033 (66)'. The 'Object Explorer' on the left shows the database structure, including tables like 'dbo.eCust', 'dbo.eTest', 'dbo.evehicle', 'dbo.Project', 'dbo.sausages', 'dbo.tBook', 'dbo.tcat', 'dbo.tcontact', and 'dbo.tcust'. The 'Columns' list for 'dbo.tcust' includes 'CustomerID (int, not null)', 'FirstName (nvarchar(50), null)', 'LastName (nvarchar(50), not null)', 'EmailAddress (nvarchar(50), null)', 'AddressLine1 (nvarchar(60), null)', 'City (nvarchar(30), null)', 'PostalCode (nvarchar(15), null)', 'CountryRegionName (nvarchar(50), null)', and 'DeleteFlag (tinyint, null)'. The 'Query Editor' on the right contains the following SQL script:

```
use jh1033
select * from tCust
where LastName = 'Smith'
```

The 'Results' pane at the bottom shows the output of the query, which consists of two rows. The first row has CustomerID 1, FirstName Georgina, LastName Smith, EmailAddress eugene10@adventure-works.com, AddressLine1 2243 W St., and City Seaford. The second row has CustomerID 501, FirstName Hany, LastName Smith, EmailAddress NULL, AddressLine1 NULL, and City London. The status bar at the bottom indicates 'Query executed successfully' and '2 rows'.

	CustomerID	FirstName	LastName	EmailAddress	AddressLine1	City
1	1	Georgina	Smith	eugene10@adventure-works.com	2243 W St.	Seaford
2	501	Hany	Smith	NULL	NULL	London

Select Query – some rows (records) – where clause using > greater than

The screenshot displays the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'cssql.jh1033 (UNIVERSITY\jh1033 (66))'. The 'Object Explorer' on the left shows the database structure, including tables like 'dbo.eTest', 'dbo.evehicle', 'dbo.Project', 'dbo.sausages', 'dbo.tBook', 'dbo.tcat', 'dbo.tcontact', 'dbo.tcust', 'dbo.tcust2', 'dbo.tcust3', 'dbo.tDepartment', 'dbo.torder', and 'dbo.torderdetail'. The 'Columns' section for 'dbo.torder' is expanded, showing fields such as 'SalesOrderID (int, not null)', 'OrderDate (date, not null)', 'DueDate (date, not null)', 'ShipDate (date, null)', 'CustomerID (int, not null)', 'SalesPersonID (int, null)', 'Status (tinyint, not null)', and 'value (money, null)'. The 'Query Editor' in the center contains the following SQL code:

```
use jh1033
select * from tOrder
where salesOrderID < 1000
```

The 'Results' pane at the bottom shows the output of the query, displaying 8 rows of data. The status bar at the bottom indicates 'Query executed successfully' and '8 rows'.

	SalesOrderID	OrderDate	DueDate	ShipDate	CustomerID	SalesPersonID	Status	value
1	1	2010-12-01	2010-12-01	2010-12-01	2	NULL	5	55.
2	2	2012-03-15	2012-03-20	2012-03-25	4	NULL	5	NL
3	2	2012-03-15	2012-03-20	2012-03-25	4	NULL	5	NL
4	3	2012-04-24	2012-04-24	2012-04-24	8	NULL	5	NL
5	7	2012-04-24	2012-04-24	2012-04-24	8	NULL	5	NL
6	7	2012-04-24	2012-04-24	2012-04-24	8	NULL	5	NL
7	10	2012-04-24	2012-04-24	2012-04-24	8	NULL	5	NL
8	11	2012-04-24	2012-04-24	2012-04-24	9	NULL	9	NL

Select Query – some rows (records) – with date range – can also use =, > (after), < (before)

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The 'Object Explorer' on the left shows the database structure for 'jh1033', including tables like 'dbo.eTest', 'dbo.evehicle', 'dbo.Project', 'dbo.sausages', 'dbo.tBook', 'dbo.tcat', 'dbo.tcontact', 'dbo.tcust', 'dbo.tcust2', 'dbo.tcust3', 'dbo.tDepartment', 'dbo.torder', 'Columns', 'Keys', 'Constraints', 'Triggers', 'Indexes', 'Statistics', 'dbo.torderdetail', 'dbo.tproduct', and 'dbo.TransactionHistory'. The 'Columns' folder is expanded, showing details for 'SalesOrderID (int, not null)', 'OrderDate (date, not null)', 'DueDate (date, not null)', 'ShipDate (date, null)', 'CustomerID (int, not null)', 'SalesPersonID (int, null)', 'Status (tinyint, not null)', and 'value (money, null)'. The 'Query Editor' window shows the following SQL query:

```
use jh1033
select * from tOrder
where orderDate between
'2003-08-01' and '2003-08-31'
```

The 'Results' tab shows the output of the query, displaying 10 rows of data. The status bar at the bottom indicates 'Query executed successfully' and '10 rows'.

	SalesOrderID	OrderDate	DueDate	ShipDate	CustomerID	SalesPersonID	Status
1	51691	2003-08-01	2003-08-13	2003-08-08	18	277	5
2	51702	2003-08-01	2003-08-13	2003-08-08	28	285	5
3	51703	2003-08-01	2003-08-13	2003-08-08	12	278	5
4	51715	2003-08-01	2003-08-13	2003-08-08	43	276	5
5	51728	2003-08-01	2003-08-13	2003-08-08	2	283	5
6	51758	2003-08-01	2003-08-13	2003-08-08	9	279	5
7	51803	2003-08-01	2003-08-13	2003-08-08	31	286	5
8	51807	2003-08-01	2003-08-13	2003-08-08	22	275	5
9	51831	2003-08-01	2003-08-13	2003-08-08	45	279	5
10	51874	2003-08-01	2003-08-13	2003-08-08	8	279	5

Select Query –selecting both fields and columns

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The 'Object Explorer' on the left shows the database structure for 'jh1033', including tables like 'dbo.eTest', 'dbo.evehicle', 'dbo.Project', 'dbo.sausages', 'dbo.tBook', 'dbo.tcat', 'dbo.tcontact', 'dbo.tcust', 'dbo.tcust2', 'dbo.tcust3', 'dbo.tDepartment', and 'dbo.torder'. The 'Columns' folder is expanded under 'dbo.torder', showing fields such as 'SalesOrderID (int, not null)', 'OrderDate (date, not null)', 'DueDate (date, not null)', 'ShipDate (date, null)', 'CustomerID (int, not null)', 'SalesPersonID (int, null)', 'Status (tinyint, not null)', and 'value (money, null)'. The 'Query' window on the right shows the following SQL query:

```
use jh1033
select dueDate, status from tOrder
where orderDate between
'2003-08-01' and '2003-08-31'
```

The 'Results' tab shows the output of the query, which is a table with two columns: 'dueDate' and 'status'. The table contains 10 rows of data, all with a status of 5 and a due date of 2003-08-13.

	dueDate	status
1	2003-08-13	5
2	2003-08-13	5
3	2003-08-13	5
4	2003-08-13	5
5	2003-08-13	5
6	2003-08-13	5
7	2003-08-13	5
8	2003-08-13	5
9	2003-08-13	5
10	2003-08-13	5

The status bar at the bottom indicates that the query was executed successfully, returning 10 rows. The status bar also shows the server name 'cssql (11.0 SP2)', the database 'UNIVERSITY\jh1033 (66)', the user 'jh1033', and the execution time '00:00:00'.

Select queries – joining tables – find matching key value

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure with tables `dbo.tOrder` and `dbo.tOrderLine`. The `Columns` and `Keys` sections are expanded for both tables. The `Columns` section for `dbo.tOrder` lists: `SalesOrderID` (PK, int, not null), `orderDate` (datetime, not null), `shipDate` (datetime, null), `custID` (int, not null), `subtotal` (money, not null), `taxAmt` (money, not null), `TotalDue` (money, not null), and `comment` (nvarchar(128), null). The `Columns` section for `dbo.tOrderLine` lists: `SalesOrderID` (PK, int, not null), `CarrierTrackingNumber` (nvarchar(255), null), `OrderQty` (smallint, not null), `ProductID` (PK, int, not null), `SpecialOfferID` (int, not null), `UnitPrice` (money, not null), `UnitPriceDiscount` (money, not null), `LineTotal` (numeric(38,6), not null), `rowguid` (uniqueidentifier, not null), and `ModifiedDate` (datetime, not null).

The central pane shows the SQL query editor with the following query:

```
select * from tOrder
inner join
tOrderLine
on tOrder.SalesOrderID = tOrderLine.SalesOrderID
order by custID
```

The bottom pane displays the query results in a table with the following columns: `SalesOrderID`, `orderDate`, `shipDate`, `custID`, `subtotal`, and `taxA`. The results show 20 rows of data, grouped by `SalesOrderID` and `custID`.

	SalesOrderID	orderDate	shipDate	custID	subtotal	taxA
1	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
2	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
3	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
4	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
5	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
6	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
7	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
8	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
9	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
10	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
11	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
12	43860	2001-08-01 00:00:00.000	2001-08-08 00:00:00.000	1	13216.0537	105
13	44501	2001-11-01 00:00:00.000	2001-11-08 00:00:00.000	1	23646.0339	189
14	44501	2001-11-01 00:00:00.000	2001-11-08 00:00:00.000	1	23646.0339	189
15	44501	2001-11-01 00:00:00.000	2001-11-08 00:00:00.000	1	23646.0339	189
16	44501	2001-11-01 00:00:00.000	2001-11-08 00:00:00.000	1	23646.0339	189
17	44501	2001-11-01 00:00:00.000	2001-11-08 00:00:00.000	1	23646.0339	189
18	44501	2001-11-01 00:00:00.000	2001-11-08 00:00:00.000	1	23646.0339	189
19	44501	2001-11-01 00:00:00.000	2001-11-08 00:00:00.000	1	23646.0339	189
20	44501	2001-11-01 00:00:00.000	2001-11-08 00:00:00.000	1	23646.0339	189

The status bar at the bottom indicates the query is ready, showing the current line (Ln 4), column (Col 26), and character (Ch 26). The status bar also shows the connection name (MOCO207-JH1033\adminjh...), the user (ci402_teaching), and the execution time (00:00:00) and row count (9,852 rows).

Putting it all together – clauses MUST go in this order

SELECT SalesOrderID, OrderDate, tOrderHeader.CustomerID,
FirstName, Lastname

Some fields

which table

FROM tOrderHeader

Join table on keys

INNER JOIN tCustomer

ON tOrderHeader.CustomerID = tCustomer.CustomerID

Limit fields

WHERE tOrderHeader.CustomerID = 11971

Order by a field
(or two!)

ORDER BY OrderDate

GROUPING and AGGREGATION

- See work from semester one – more complex than this – can review as required.

Other work

- Review detail on SQL statements from Study Materials
- Review grouping and aggregation material from MyStudies
- Review *Common Database Errors and How To Resolve Them* (Study materials)
- Use these online databases to practice skills at home:
 - https://sqlbolt.com/lesson/select_queries_introduction
 - <http://www.sqlcourse.com/>
 - <http://www.w3schools.com/sql/>
 - <https://www.codecademy.com/learn/learn-sql>