SQL: Insert, Update and Delete

- Management Studio provides user interface:
  - inserts, updates and deletes
  - demo this feature
- There is SQL for these tasks – you'd use this through an app etc.
- Note – use your own database!!
  - You do not have rights to change other people's data

SELECT INTO

```
select * INTO tcust2
     from tcust                    //sqlServer

create table tCust2
As select * from tCust             //mySql
```

Run query1

This creates a new table – tcust2

…. **generates an error if table already exists.**

(We're going to mess around with this new table and delete it at the end)

tCust - fields

custID

addressLine1

addressLine2

fName

sName

city

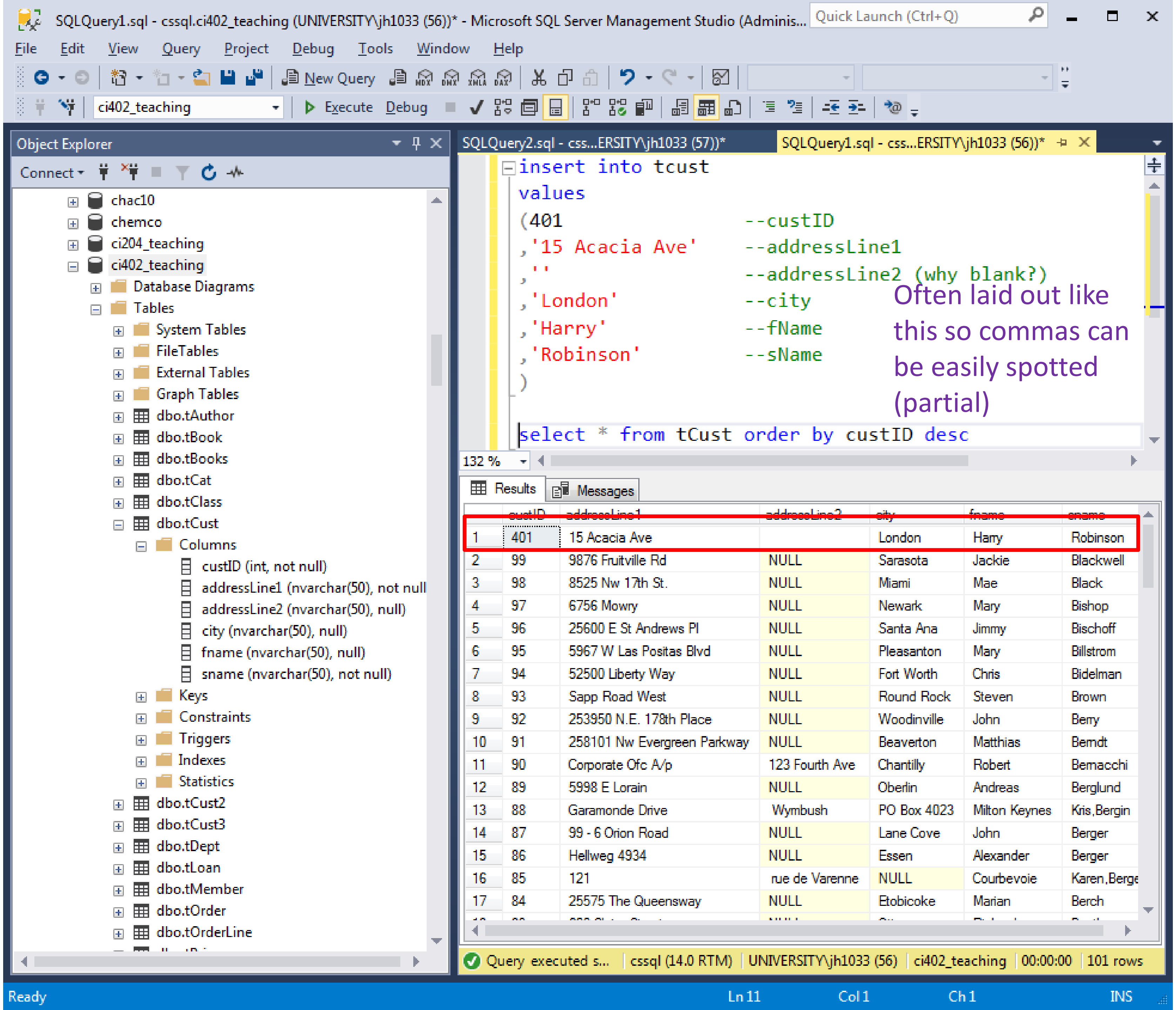
INSERT

```
insert into tCust2
/*already made a copy of tCust using select into (q1)*/
values
(401                                /*custID*/
, 'Harry'                           /*fName*/
, 'Robinson'                        /*sName*/
, '15 Acacia Ave'                   /*addressLine1*/
, ''                                /*addressLine2 (why blank?)*/
, 'London'                          /*city*/
, 'UK'                              /*country*/
, '1'                               /*delete flag*/
, 'B'                               /*customer category*/
)
```

run query2 (watch out for the quote marks – esp. if you copy from here – need to be NOT SMART QUOTES *?stupid quotes*)

If you have values for **all** the fields then no need to identify the fields (bad practice).

Values must be in the same order as columns (use ObjectExplorer/ phpMyAdmin left-most panel) to see them)


SQLQuery1.sql - cssql.ci402_teaching (UNIVERSITY\jh1033 (56))* - Microsoft SQL Server Management Studio (Adminis...
Quick Launch (Ctrl+Q)
File Edit View Query Project Debug Tools Window Help
New Query
ci402_teaching
Execute Debug
Object Explorer
Connect
SQLQuery2.sql - css...ERSITY\jh1033 (57))*
SQLQuery1.sql - css...ERSITY\jh1033 (56))*
132 %
Results
Messages
Ready
Ln 11
Col 1
Ch 1
INS

```
chac10
chemco
ci204_teaching
ci402_teaching
    Database Diagrams
    Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.tAuthor
        dbo.tBook
        dbo.tBooks
        dbo.tCat
        dbo.tClass
        dbo.tCust
            Columns
                custID (int, not null)
                addressLine1 (nvarchar(50), not null
                addressLine2 (nvarchar(50), null)
                city (nvarchar(50), null)
                fname (nvarchar(50), null)
                sname (nvarchar(50), not null)
            Keys
            Constraints
            Triggers
            Indexes
            Statistics
        dbo.tCust2
        dbo.tCust3
        dbo.tDept
        dbo.tLoan
        dbo.tMember
        dbo.tOrder
        dbo.tOrderLine
```

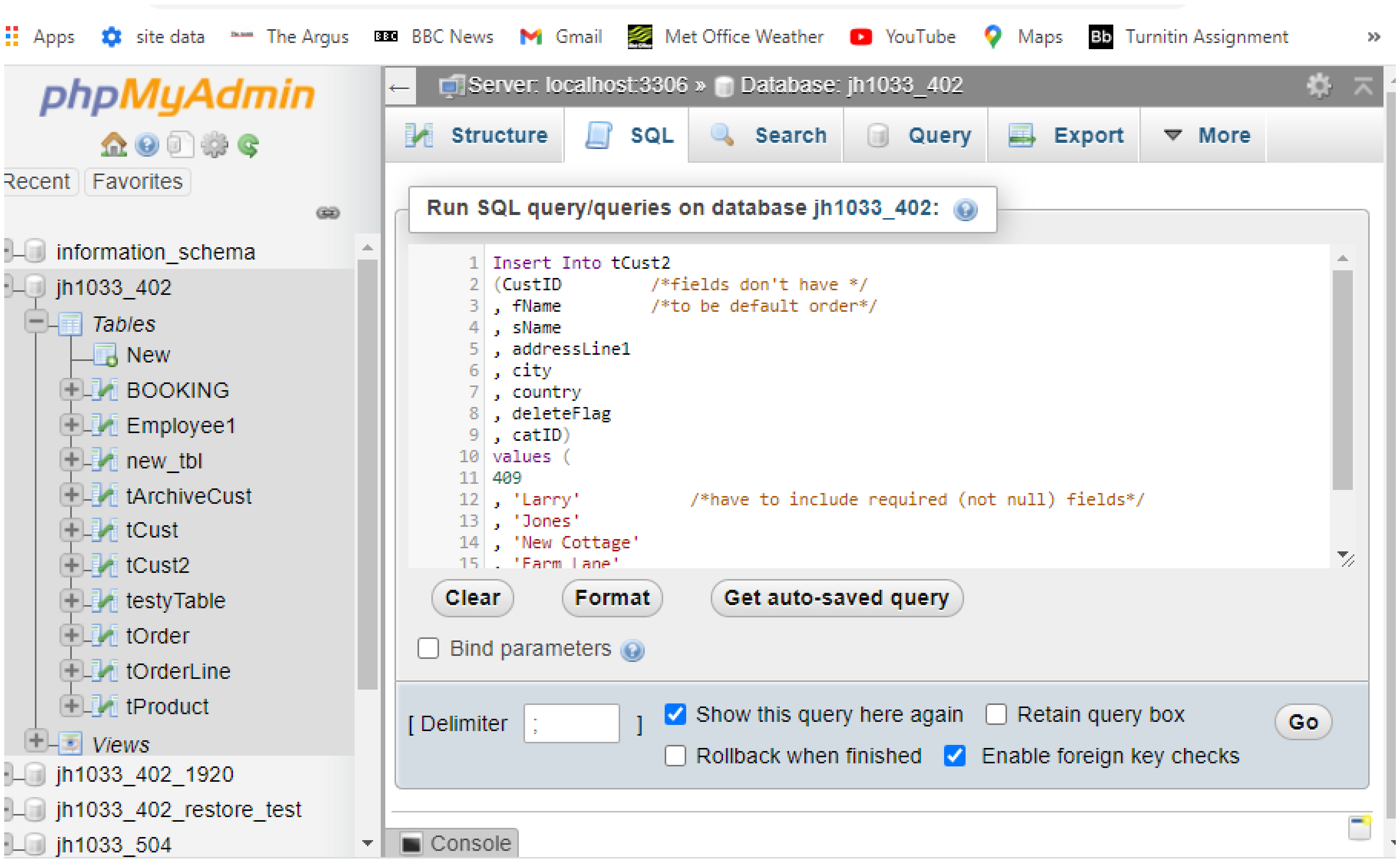
```sql
insert into tcust
values
(401                --custID
,'15 Acacia Ave'    --addressLine1
,''                 --addressLine2 (why blank?)
,'London'           --city
,'Harry'            --fName
,'Robinson'         --sName
)

select * from tCust order by custID desc
```

Often laid out like this so commas can be easily spotted (partial)

	custID	addressLine1	addressLine2	city	fname	sname
1	401	15 Acacia Ave		London	Harry	Robinson
2	99	9876 Fruitville Rd	NULL	Sarasota	Jackie	Blackwell
3	98	8525 Nw 17th St.	NULL	Miami	Mae	Black
4	97	6756 Mowry	NULL	Newark	Mary	Bishop
5	96	25600 E St Andrews Pl	NULL	Santa Ana	Jimmy	Bischoff
6	95	5967 W Las Positas Blvd	NULL	Pleasanton	Mary	Billstrom
7	94	52500 Liberty Way	NULL	Fort Worth	Chris	Bidelman
8	93	Sapp Road West	NULL	Round Rock	Steven	Brown
9	92	253950 N.E. 178th Place	NULL	Woodinville	John	Berry
10	91	258101 Nw Evergreen Parkway	NULL	Beaverton	Matthias	Berndt
11	90	Corporate Ofc A/p	123 Fourth Ave	Chantilly	Robert	Bernacchi
12	89	5998 E Lorain	NULL	Oberlin	Andreas	Berglund
13	88	Garamonde Drive	Wymbush	PO Box 4023	Milton Keynes	Kris,Bergin
14	87	99 - 6 Orion Road	NULL	Lane Cove	John	Berger
15	86	Hellweg 4934	NULL	Essen	Alexander	Berger
16	85	121	rue de Varenne	NULL	Courbevoie	Karen,Berge
17	84	25575 The Queensway	NULL	Etobicoke	Marian	Berch

Query executed s... | cssql (14.0 RTM) | UNIVERSITY\jh1033 (56) | ci402_teaching | 00:00:00 | 101 rows



Run SQL query/queries on database jh1033_402:

```
Insert Into tCust2
(CustID         /*fields don't have */
, fName         /*to be default order*/
, sName
, addressLine1
, city
, country
, deleteFlag
, catID)
values (
409
, 'Larry'           /*have to include required (not null) fields*/
, 'Jones'
, 'New Cottage'
, 'Farm Lane'
```

Bind parameters: ☐

[Delimiter ;]

☑ Show this query here again
☐ Retain query box
☐ Rollback when finished
☑ Enable foreign key checks

INSERT subset of fields

```
insert into tCust
(CustID                          --fields don't have
, fName                          --to be default order
, sName
, addressLine1
, city
, country
, deleteFlag
, catID)
values (
   406
, 'Larry'                        --have to include required (not null) fields
, 'Jones’                        --no address line 2 here as it’s optional field
, 'New Cottage'
, 'Farm Lane‛
, 'Brighton'
, 1
, 'S')
```

run query2a Notice you have to identify *each* field. The values must be in the correct order, matching the field listing

Using SELECT with INSERT

```
INSERT INTO tCust2
      SELECT * FROM tCust    //both systems

      /*needs table to already exist
      --can run again and again (as long as doesn't
      --break primary key rules)
      --how many rows in tCust2 after this query
      runs?*/
--query3
```

- needs table to already exist
- can run again and again (as long as doesn't break primary key rules – *can't after amend tCust2 as new field – next q*)

CHANGE TABLE

```
ALTER TABLE tCust2
ADD postalCode varchar(7)
```

Run query4

Adds a new column (postalCode) to tCust2.

What’s in the new column?

Sql

Table / field

UPDATE

- Update one record:

```
UPDATE tcust2
SET fName = 'Harry'
WHERE custid = 1   /*why use custID, why not name?*/
```

[Query5]

Sql
Table / field
value
comment

Update multiple records

```
UPDATE tCust2
SET city = 'Miami FLA'
WHERE city = 'Miami'
```

[Query6]

Sql

Table / field

value

Update multiple fields

```
UPDATE tCust2                  /*table to change*/
SET PostalCode = 'M4'          /*field(s) to change*/
, city = 'Toronto, Ontario'
WHERE city = 'Toronto'         /*identifier*/
```

Query7

Sql
Table / field
value
comment

Delete

- Delete from tCust2
  - ¡¡¡ This SQL deletes all the records in the table!!!
  - That's why I haven't put it as a pre-made query
  - Notice no field names specified
  - You delete the **whole record** not a part

Delete from where

Run query8 to create tcust3 then query9 to see how many Portlandians there are

```
Delete from tCust3
where city = 'Portland'
```

- Run query10

This deletes any Portland records

Sql
Table / field
value
comment

Virtual Delete

- Database Administrators avoid *real* deletes where possible

- **Virtual deletes** use **views** and **filtering** to create a virtual delete without an actual delete of the record

Virtual Delete - DeleteFlag

- Add a delete flag to the table
  - DeleteFlag - datatype bit 1 or 0
  - datatype boolean - TRUE or FALSE

query11

Virtual Delete - View

```
create view vtCust
AS
Select *
from tCust
where deleteflag = 0  //i.e. not deleted
```

query11a

Sql
Table / field
value
comment

Test the view with *select * from vtCust*
(can treat a VIEW just like a table)
query12

Update DeleteFlag

- Update all the Salt Lake City records so that DeleteFlag is TRUE (2 rows)

Run query13

- `Select * from vtCust;`
- `Select * from tCust3; --compare the outcomes`

- `Or use count(*)`

Applications use virtual delete

- Users press the delete button
- Update SQL runs – sets *DeleteFlag* to *TRUE (boolean)* or *1 (bit)*
- Page is refreshed from VIEW
- Updated record(s) appear to have been deleted

Archiving

- If table becomes too large
  - i.e. performance is affected

- Archive “deleted” records
- Copy “deleted” records to an archive table

```
Select * into tArchiveCust
From tCust
Where DeleteFlag = 1
```

Query 11a (slightly diff. syntax for MySQL included)