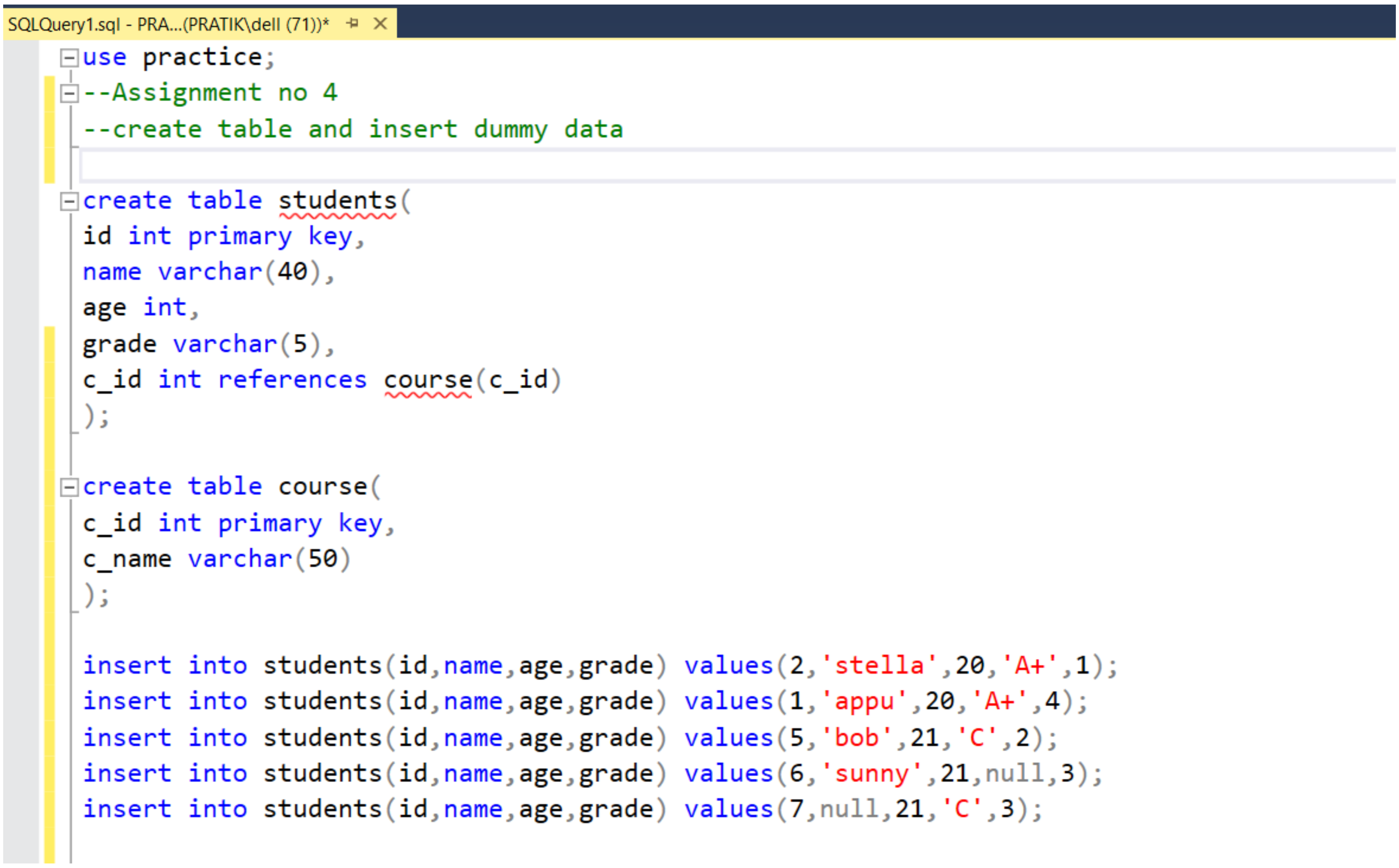
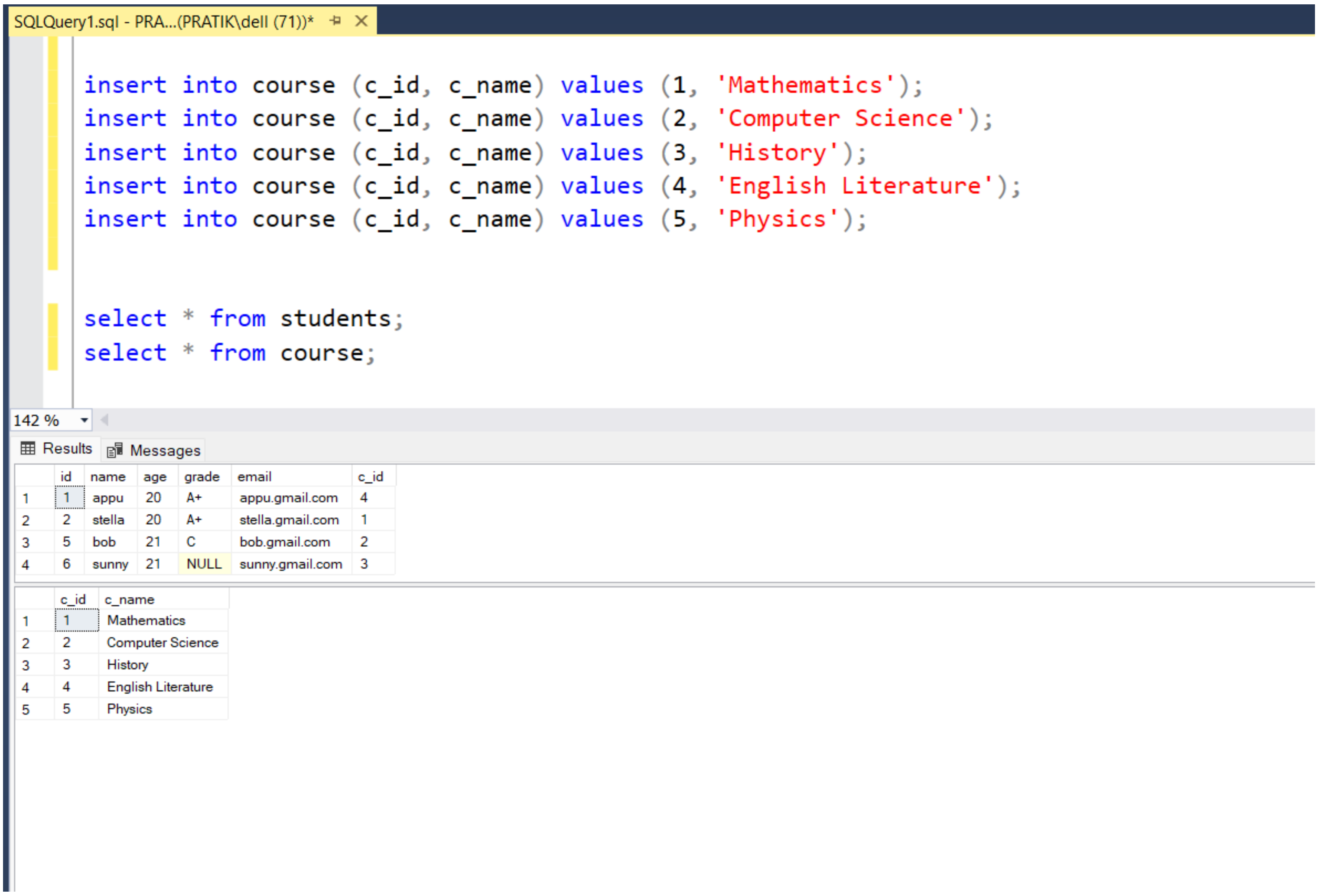
# Assignment No: 5

Name: Pratik Wani

- SQL Queries
    - Here I Used the database named as ‘Practice’ and created table named as ‘Students’ and ‘Courses’ to perform exercises.



```sql
use practice;
--Assignment no 4
--create table and insert dummy data

create table students(
id int primary key,
name varchar(40),
age int,
grade varchar(5),
c_id int references course(c_id)
);

create table course(
c_id int primary key,
c_name varchar(50)
);

insert into students(id,name,age,grade) values(2,'stella',20,'A+',1);
insert into students(id,name,age,grade) values(1,'appu',20,'A+',4);
insert into students(id,name,age,grade) values(5,'bob',21,'C',2);
insert into students(id,name,age,grade) values(6,'sunny',21,null,3);
insert into students(id,name,age,grade) values(7,null,21,'C',3);
```



SQLQuery1.sql - PRA...(PRATIK\dell (71))*

```sql
insert into course (c_id, c_name) values (1, 'Mathematics');
insert into course (c_id, c_name) values (2, 'Computer Science');
insert into course (c_id, c_name) values (3, 'History');
insert into course (c_id, c_name) values (4, 'English Literature');
insert into course (c_id, c_name) values (5, 'Physics');


select * from students;
select * from course;
```

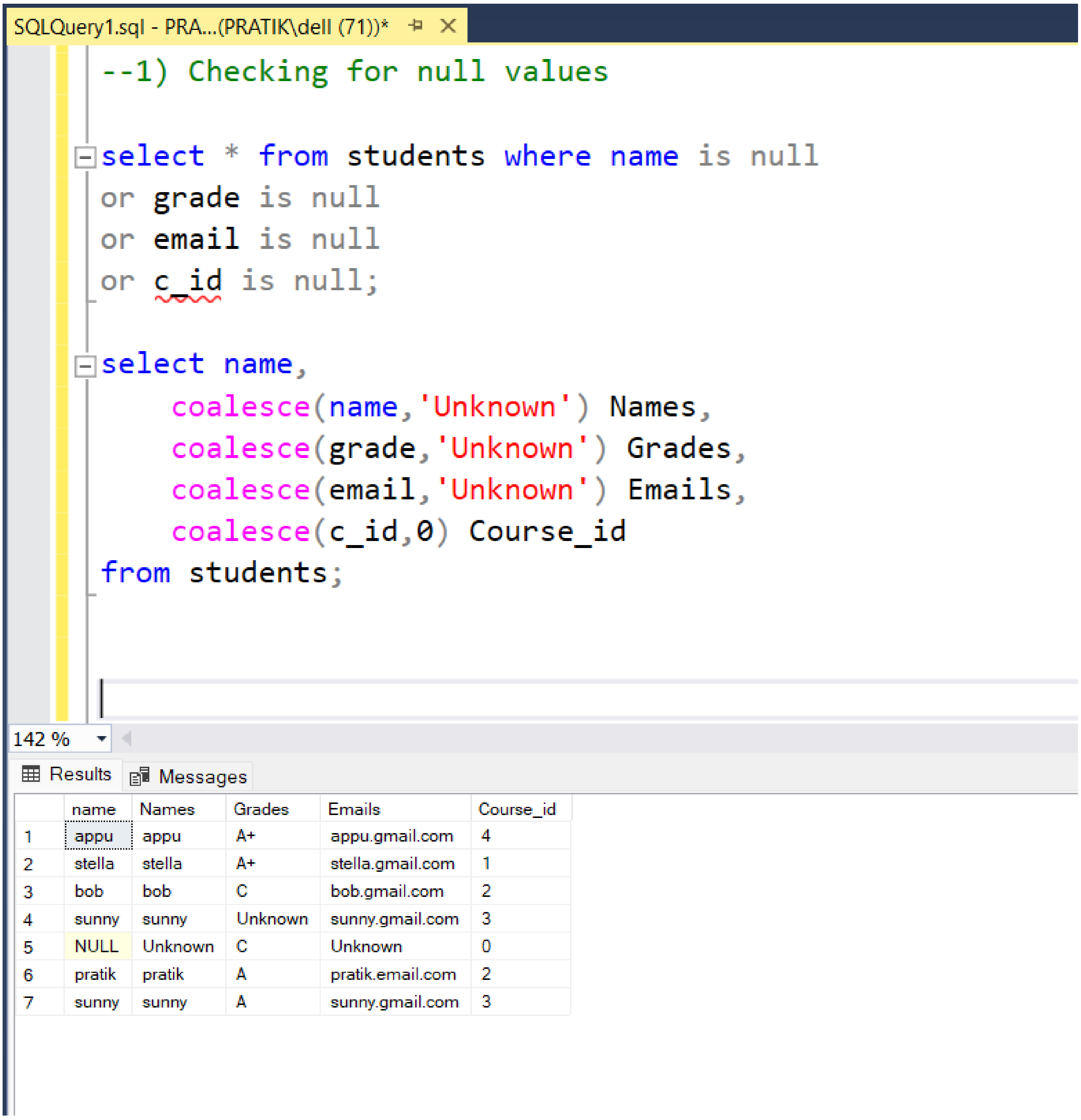
142 %

Results | Messages

|   | id | name | age | grade | email | c_id |
|---|---|---|---|---|---|---|
| 1 | 1 | appu | 20 | A+ | appu.gmail.com | 4 |
| 2 | 2 | stella | 20 | A+ | stella.gmail.com | 1 |
| 3 | 5 | bob | 21 | C | bob.gmail.com | 2 |
| 4 | 6 | sunny | 21 | NULL | sunny.gmail.com | 3 |

|   | c_id | c_name |
|---|---|---|
| 1 | 1 | Mathematics |
| 2 | 2 | Computer Science |
| 3 | 3 | History |
| 4 | 4 | English Literature |
| 5 | 5 | Physics |

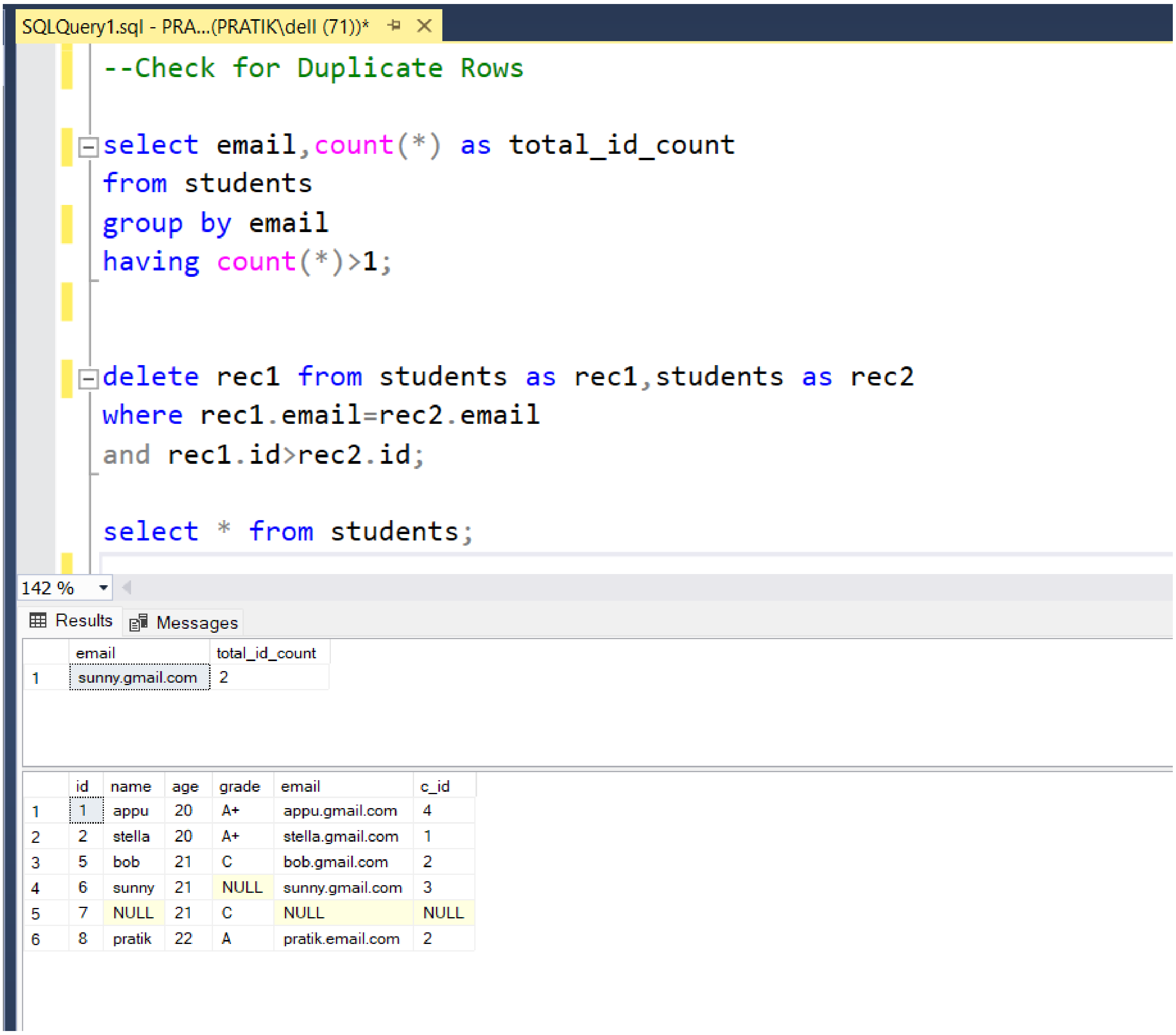
- <u>Data Cleansing and Transformation:</u>
  - Check for Missing Values:



```sql
--1) Checking for null values

select * from students where name is null
or grade is null
or email is null
or c_id is null;

select name,
    coalesce(name,'Unknown') Names,
    coalesce(grade,'Unknown') Grades,
    coalesce(email,'Unknown') Emails,
    coalesce(c_id,0) Course_id
from students;
```

|   | name | Names | Grades | Emails | Course_id |
|---|---|---|---|---|---|
| 1 | appu | appu | A+ | appu.gmail.com | 4 |
| 2 | stella | stella | A+ | stella.gmail.com | 1 |
| 3 | bob | bob | C | bob.gmail.com | 2 |
| 4 | sunny | sunny | Unknown | sunny.gmail.com | 3 |
| 5 | NULL | Unknown | C | Unknown | 0 |
| 6 | pratik | pratik | A | pratik.email.com | 2 |
| 7 | sunny | sunny | A | sunny.gmail.com | 3 |

- Check for Duplicate Rows:



SQLQuery1.sql - PRA...(PRATIK\dell (71))*

```sql
--Check for Duplicate Rows

select email,count(*) as total_id_count
from students
group by email
having count(*)>1;


delete rec1 from students as rec1,students as rec2
where rec1.email=rec2.email
and rec1.id>rec2.id;

select * from students;
```

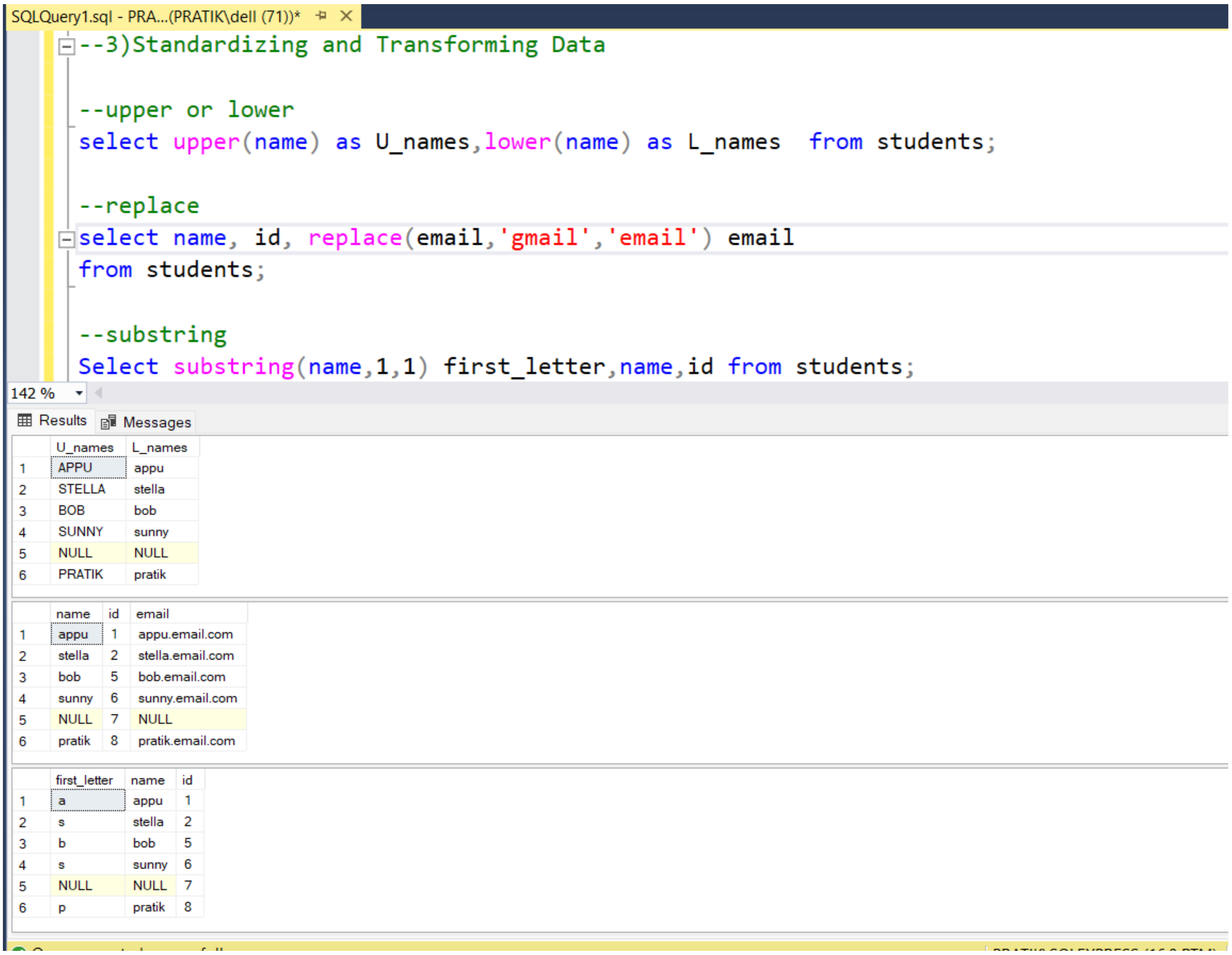
142 %

Results | Messages

| | email | total_id_count |
|---|---|---|
| 1 | sunny.gmail.com | 2 |

| | id | name | age | grade | email | c_id |
|---|---|---|---|---|---|---|
| 1 | 1 | appu | 20 | A+ | appu.gmail.com | 4 |
| 2 | 2 | stella | 20 | A+ | stella.gmail.com | 1 |
| 3 | 5 | bob | 21 | C | bob.gmail.com | 2 |
| 4 | 6 | sunny | 21 | NULL | sunny.gmail.com | 3 |
| 5 | 7 | NULL | 21 | C | NULL | NULL |
| 6 | 8 | pratik | 22 | A | pratik.email.com | 2 |

- Standardizing and Transforming Data:
  - Replace
  - Upper or Lower
  - Substring
  - Trim
  - Date Format
  - Use Case



SQLQuery1.sql - PRA...(PRATIK\dell (71))*

```
--3)Standardizing and Transforming Data

--upper or lower
select upper(name) as U_names,lower(name) as L_names  from students;

--replace
select name, id, replace(email,'gmail','email') email
from students;

--substring
Select substring(name,1,1) first_letter,name,id from students;
```

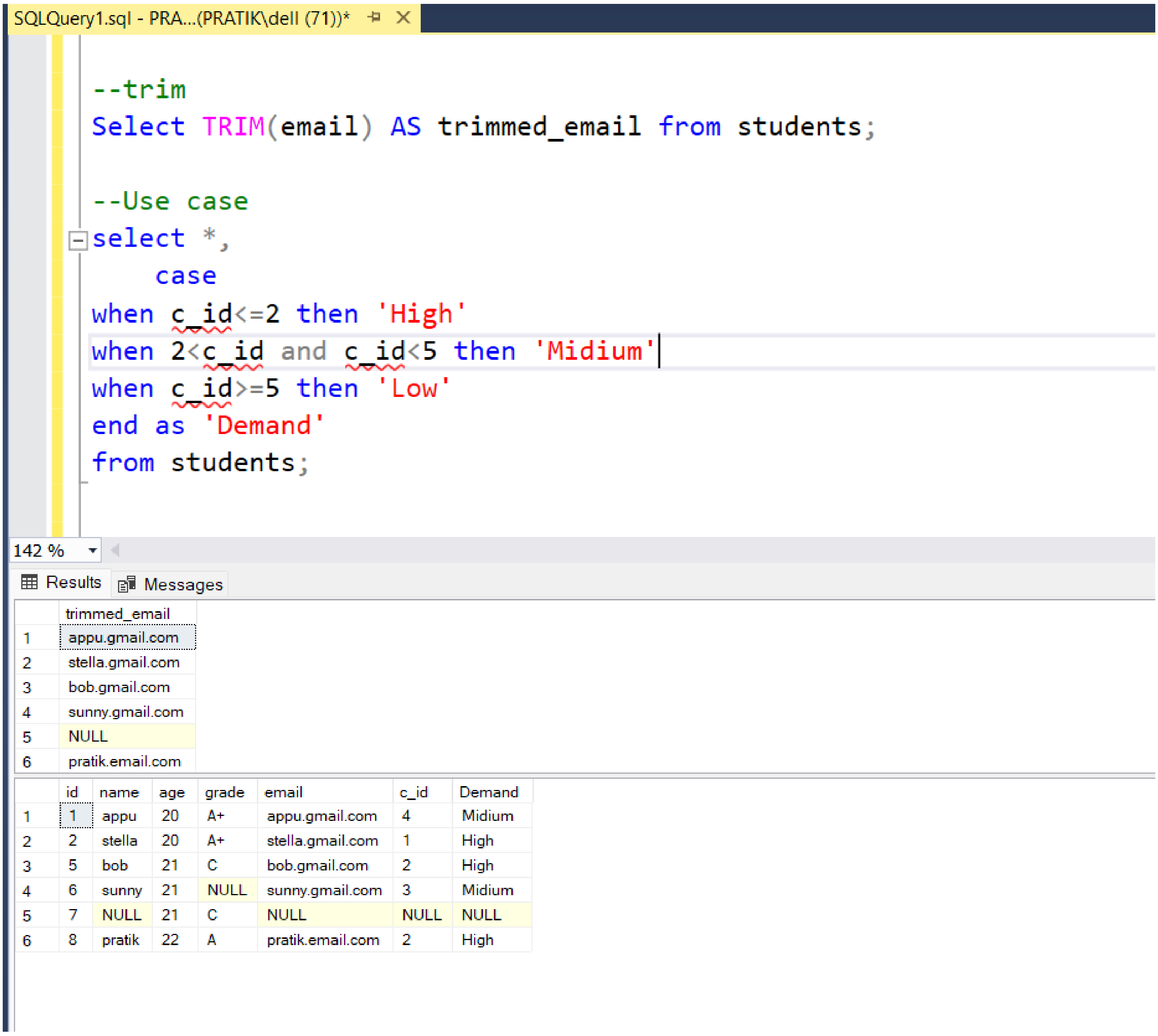
142 %

Results | Messages

| | U_names | L_names |
|---|---|---|
| 1 | APPU | appu |
| 2 | STELLA | stella |
| 3 | BOB | bob |
| 4 | SUNNY | sunny |
| 5 | NULL | NULL |
| 6 | PRATIK | pratik |

| | name | id | email |
|---|---|---|---|
| 1 | appu | 1 | appu.email.com |
| 2 | stella | 2 | stella.email.com |
| 3 | bob | 5 | bob.email.com |
| 4 | sunny | 6 | sunny.email.com |
| 5 | NULL | 7 | NULL |
| 6 | pratik | 8 | pratik.email.com |

| | first_letter | name | id |
|---|---|---|---|
| 1 | a | appu | 1 |
| 2 | s | stella | 2 |
| 3 | b | bob | 5 |
| 4 | s | sunny | 6 |
| 5 | NULL | NULL | 7 |
| 6 | p | pratik | 8 |



SQLQuery1.sql - PRA...(PRATIK\dell (71))*

```sql
--trim
Select TRIM(email) AS trimmed_email from students;

--Use case
select *,
    case
when c_id<=2 then 'High'
when 2<c_id and c_id<5 then 'Midium'
when c_id>=5 then 'Low'
end as 'Demand'
from students;
```

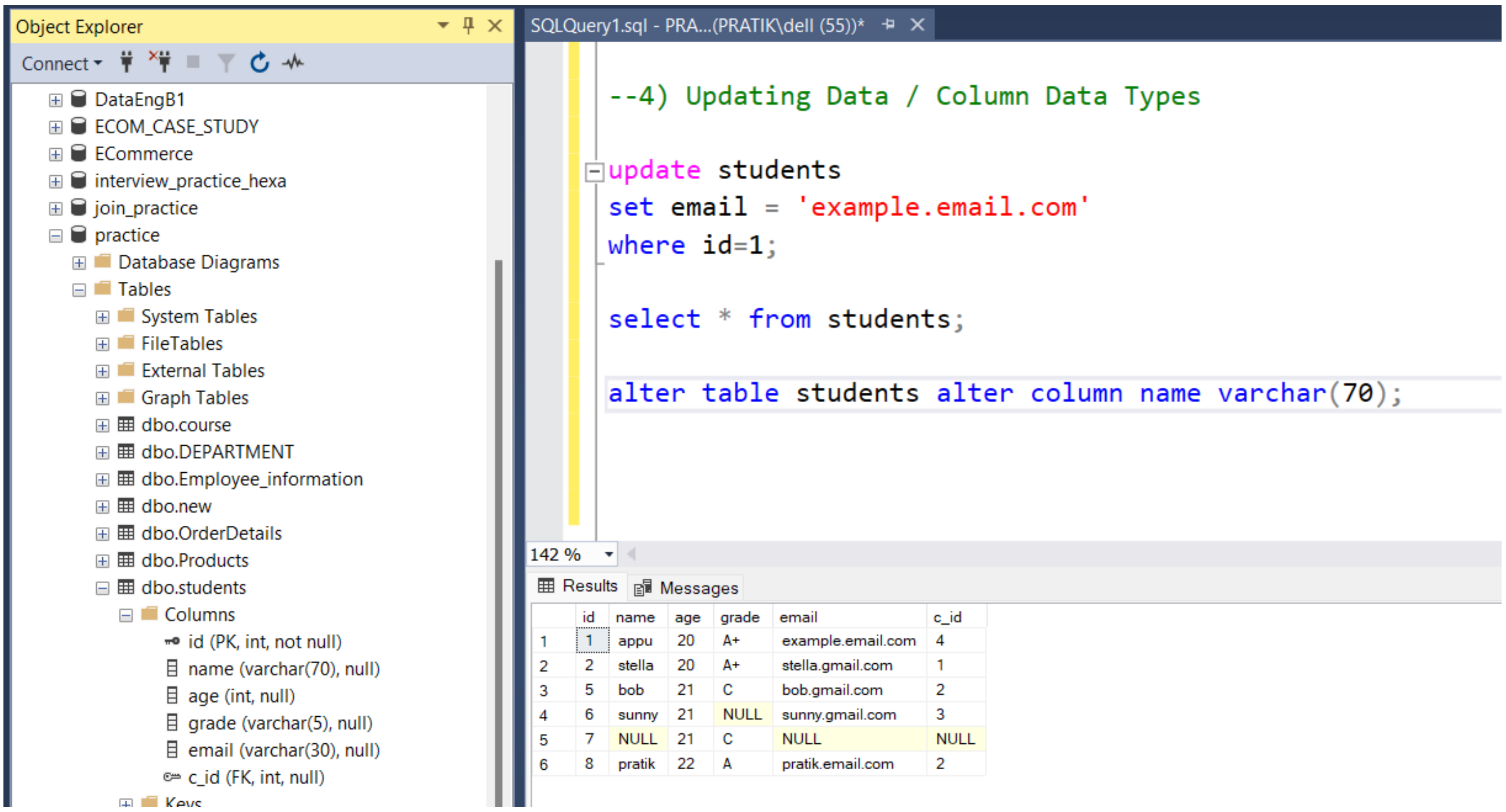
142 %

Results | Messages

| | trimmed_email |
|---|---|
| 1 | appu.gmail.com |
| 2 | stella.gmail.com |
| 3 | bob.gmail.com |
| 4 | sunny.gmail.com |
| 5 | NULL |
| 6 | pratik.email.com |

| | id | name | age | grade | email | c_id | Demand |
|---|---|---|---|---|---|---|---|
| 1 | 1 | appu | 20 | A+ | appu.gmail.com | 4 | Midium |
| 2 | 2 | stella | 20 | A+ | stella.gmail.com | 1 | High |
| 3 | 5 | bob | 21 | C | bob.gmail.com | 2 | High |
| 4 | 6 | sunny | 21 | NULL | sunny.gmail.com | 3 | Midium |
| 5 | 7 | NULL | 21 | C | NULL | NULL | NULL |
| 6 | 8 | pratik | 22 | A | pratik.email.com | 2 | High |

- Updating Data / Column Data Types:

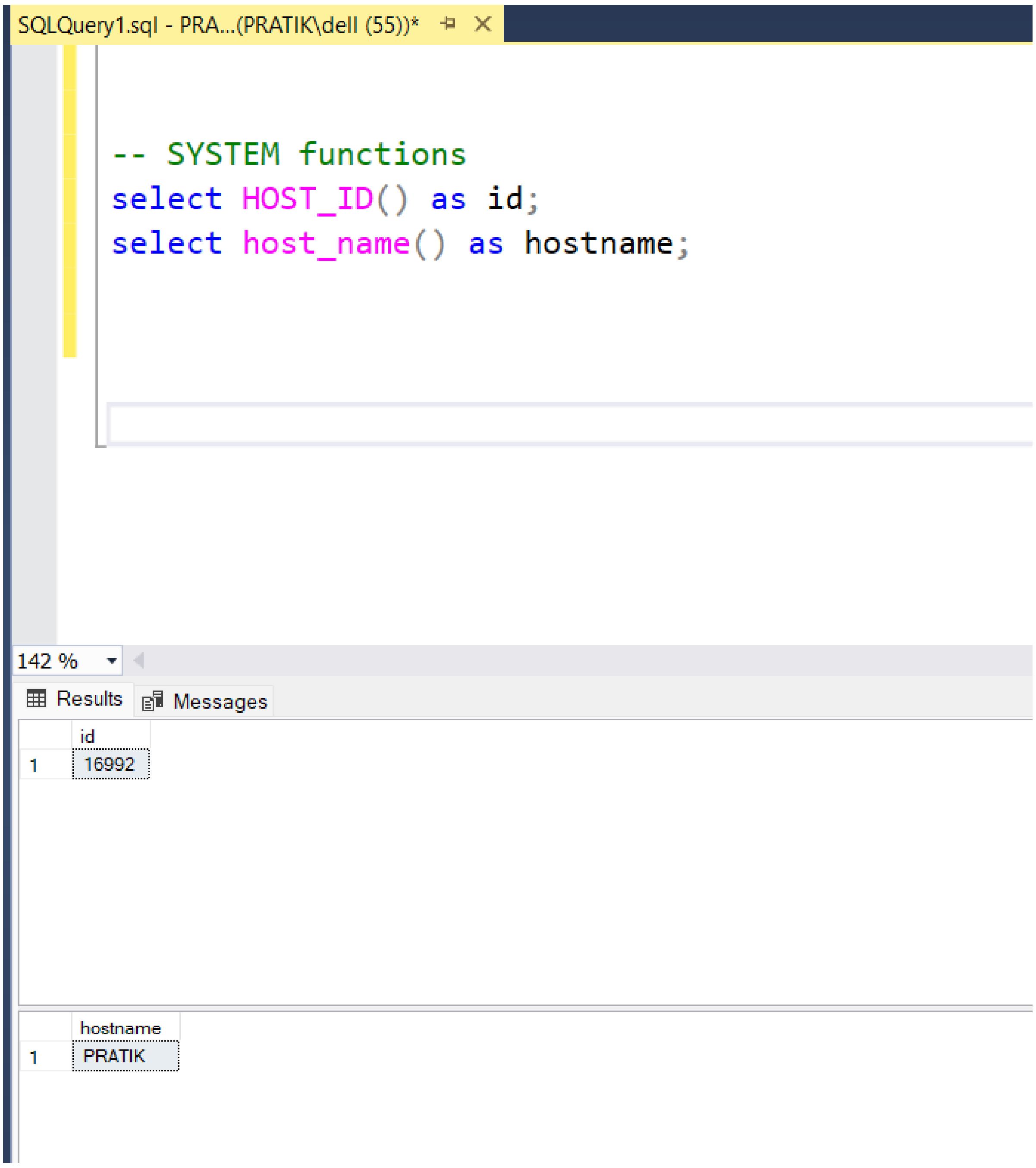


- DataEngB1
- ECOM_CASE_STUDY
- ECommerce
- interview_practice_hexa
- join_practice
- practice
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - External Tables
    - Graph Tables
    - dbo.course
    - dbo.DEPARTMENT
    - dbo.Employee_information
    - dbo.new
    - dbo.OrderDetails
    - dbo.Products
    - dbo.students
      - Columns
        - id (PK, int, not null)
        - name (varchar(70), null)
        - age (int, null)
        - grade (varchar(5), null)
        - email (varchar(30), null)
        - c_id (FK, int, null)
      - Keys

```sql
--4) Updating Data / Column Data Types

update students
set email = 'example.email.com'
where id=1;

select * from students;

alter table students alter column name varchar(70);
```

| | id | name | age | grade | email | c_id |
|---|---|---|---|---|---|---|
| 1 | 1 | appu | 20 | A+ | example.email.com | 4 |
| 2 | 2 | stella | 20 | A+ | stella.gmail.com | 1 |
| 3 | 5 | bob | 21 | C | bob.gmail.com | 2 |
| 4 | 6 | sunny | 21 | NULL | sunny.gmail.com | 3 |
| 5 | 7 | NULL | 21 | C | NULL | NULL |
| 6 | 8 | pratik | 22 | A | pratik.email.com | 2 |

- System Function:

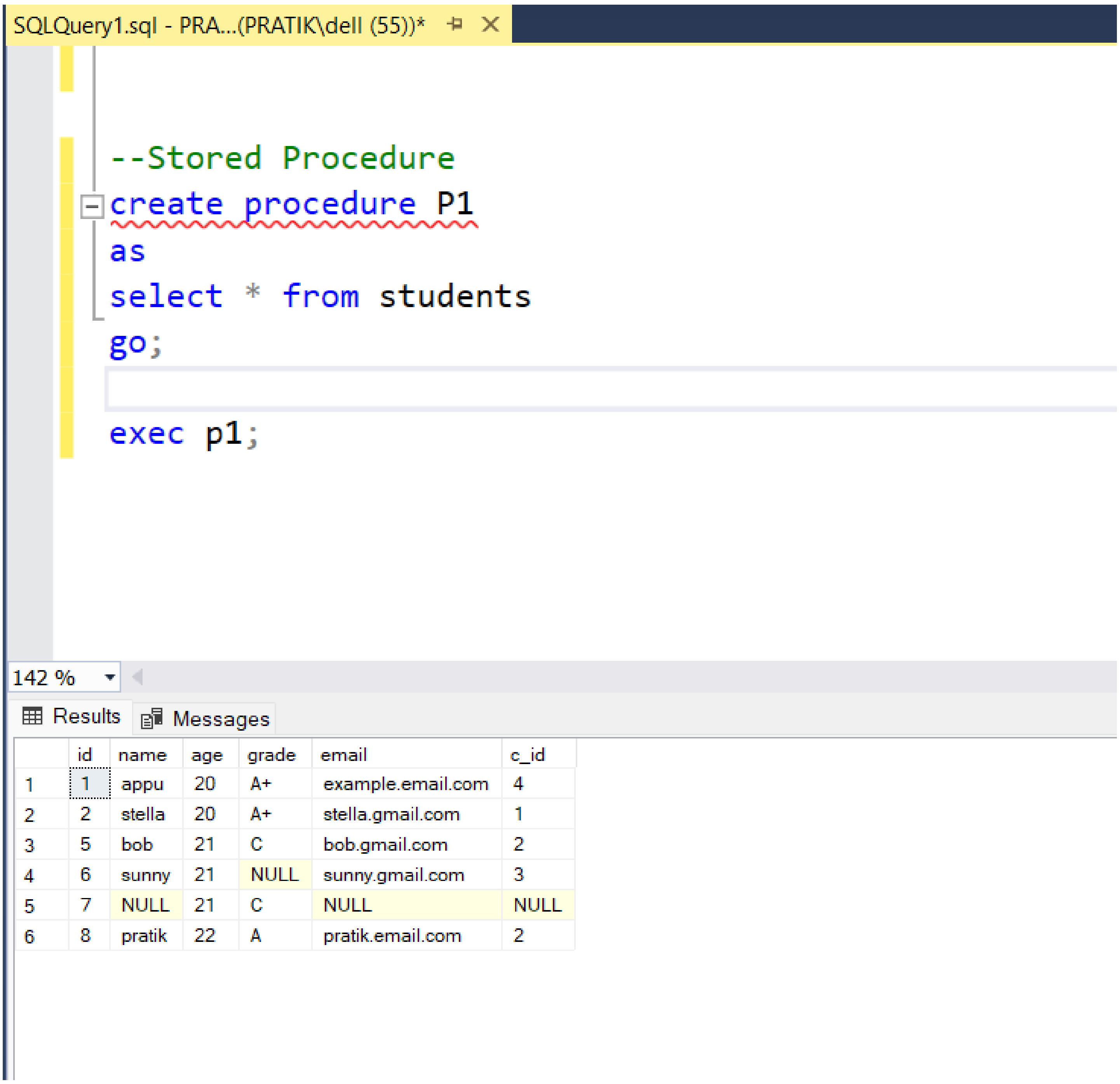


```
-- SYSTEM functions
select HOST_ID() as id;
select host_name() as hostname;
```

```
   id
1  16992
```

```
   hostname
1  PRATIK
```

- Stored Procedure:



SQLQuery1.sql - PRA...(PRATIK\dell (55))*

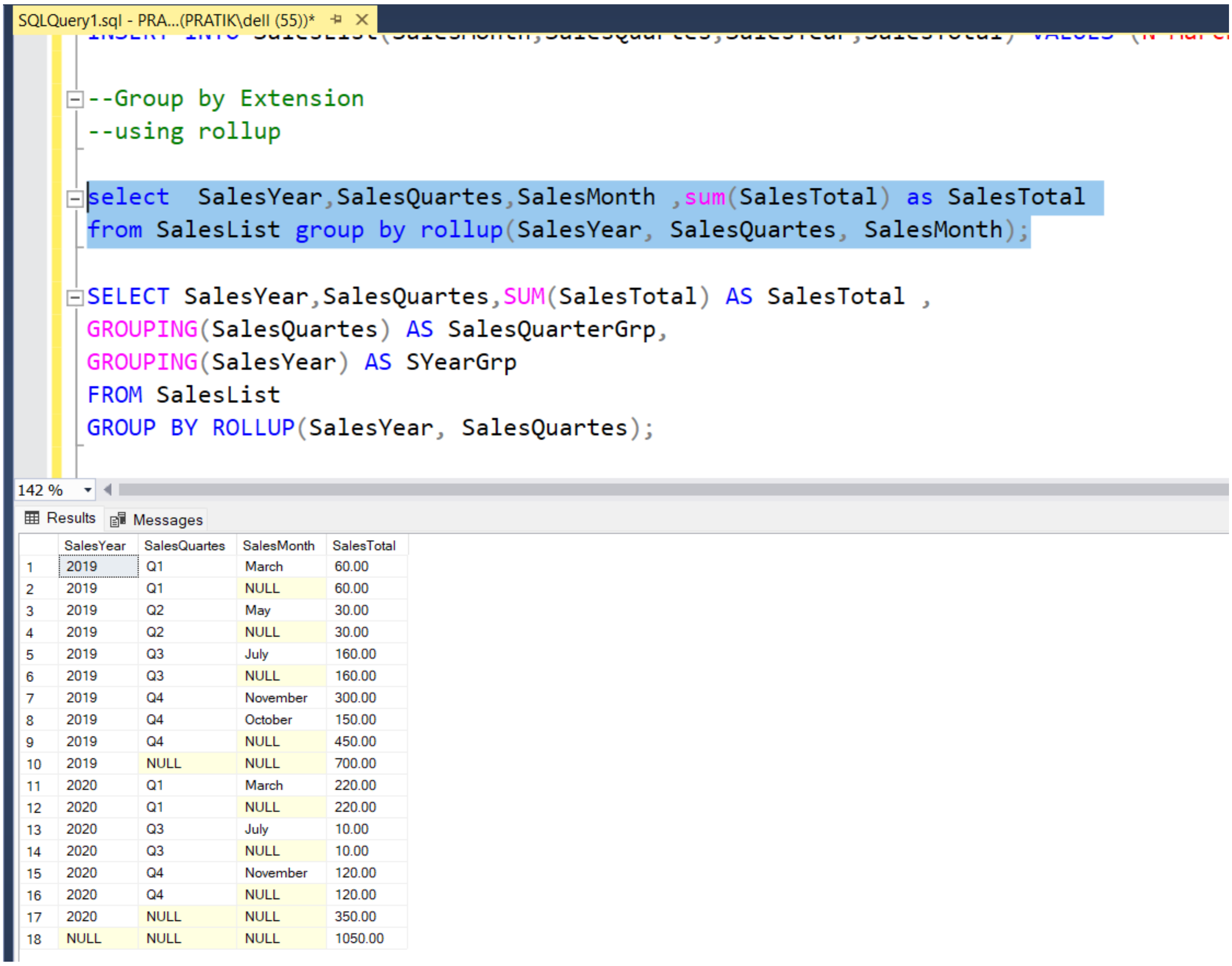
```
--Stored Procedure
create procedure P1
as
select * from students
go;

exec p1;
```

142 %

Results | Messages

```
   id  name    age  grade  email               c_id
1  1   appu    20   A+     example.email.com   4
2  2   stella  20   A+     stella.gmail.com    1
3  5   bob     21   C      bob.gmail.com       2
4  6   sunny   21   NULL   sunny.gmail.com     3
5  7   NULL    21   C      NULL                NULL
6  8   pratik  22   A      pratik.email.com    2
```

- <u>Group by Extension(Rollup) and Grouping</u>:



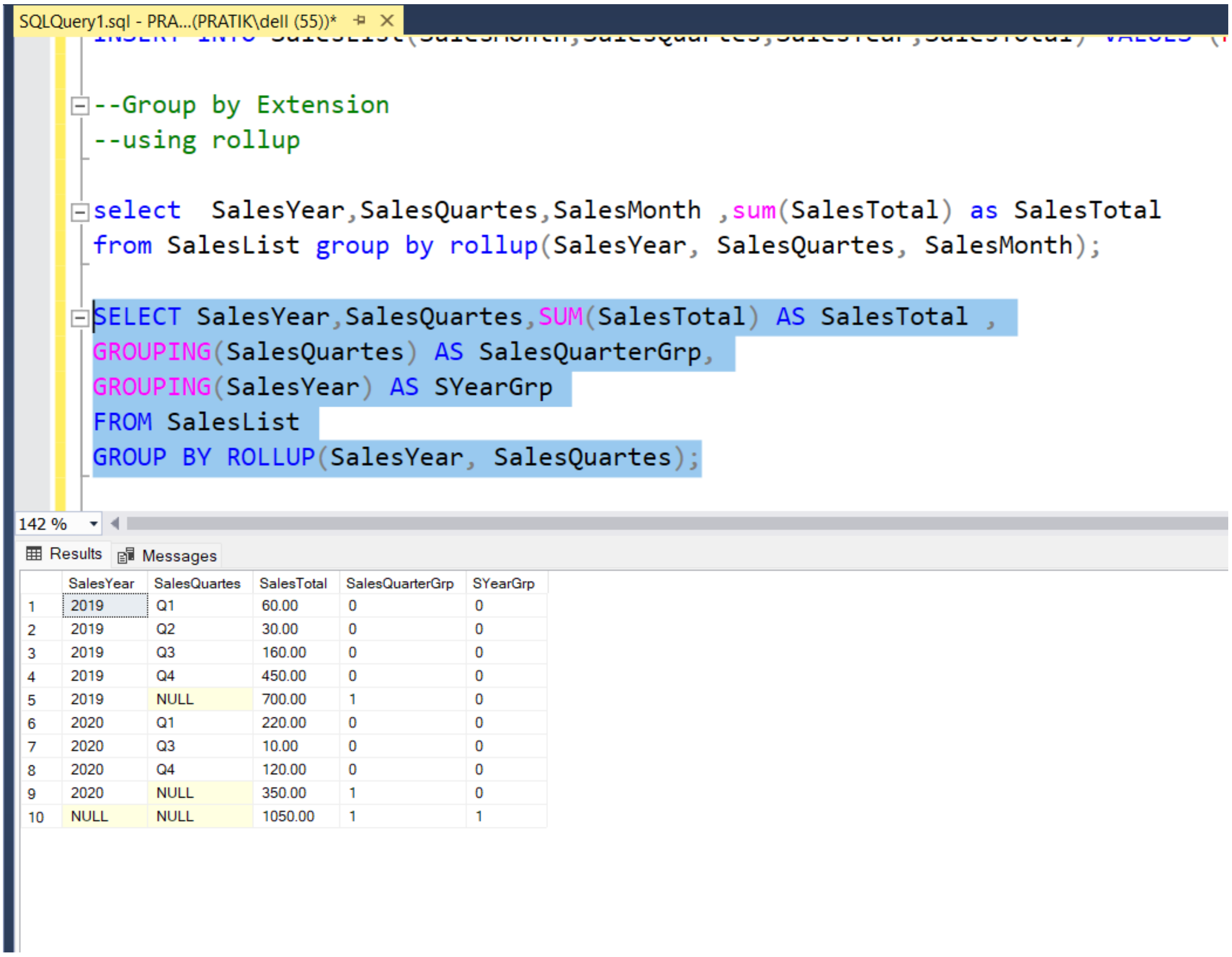
SQLQuery1.sql - PRA...(PRATIK\dell (55))*

```
--Group by Extension
--using rollup

select  SalesYear,SalesQuartes,SalesMonth ,sum(SalesTotal) as SalesTotal
from SalesList group by rollup(SalesYear, SalesQuartes, SalesMonth);

SELECT SalesYear,SalesQuartes,SUM(SalesTotal) AS SalesTotal ,
GROUPING(SalesQuartes) AS SalesQuarterGrp,
GROUPING(SalesYear) AS SYearGrp
FROM SalesList
GROUP BY ROLLUP(SalesYear, SalesQuartes);
```

142 %

Results | Messages

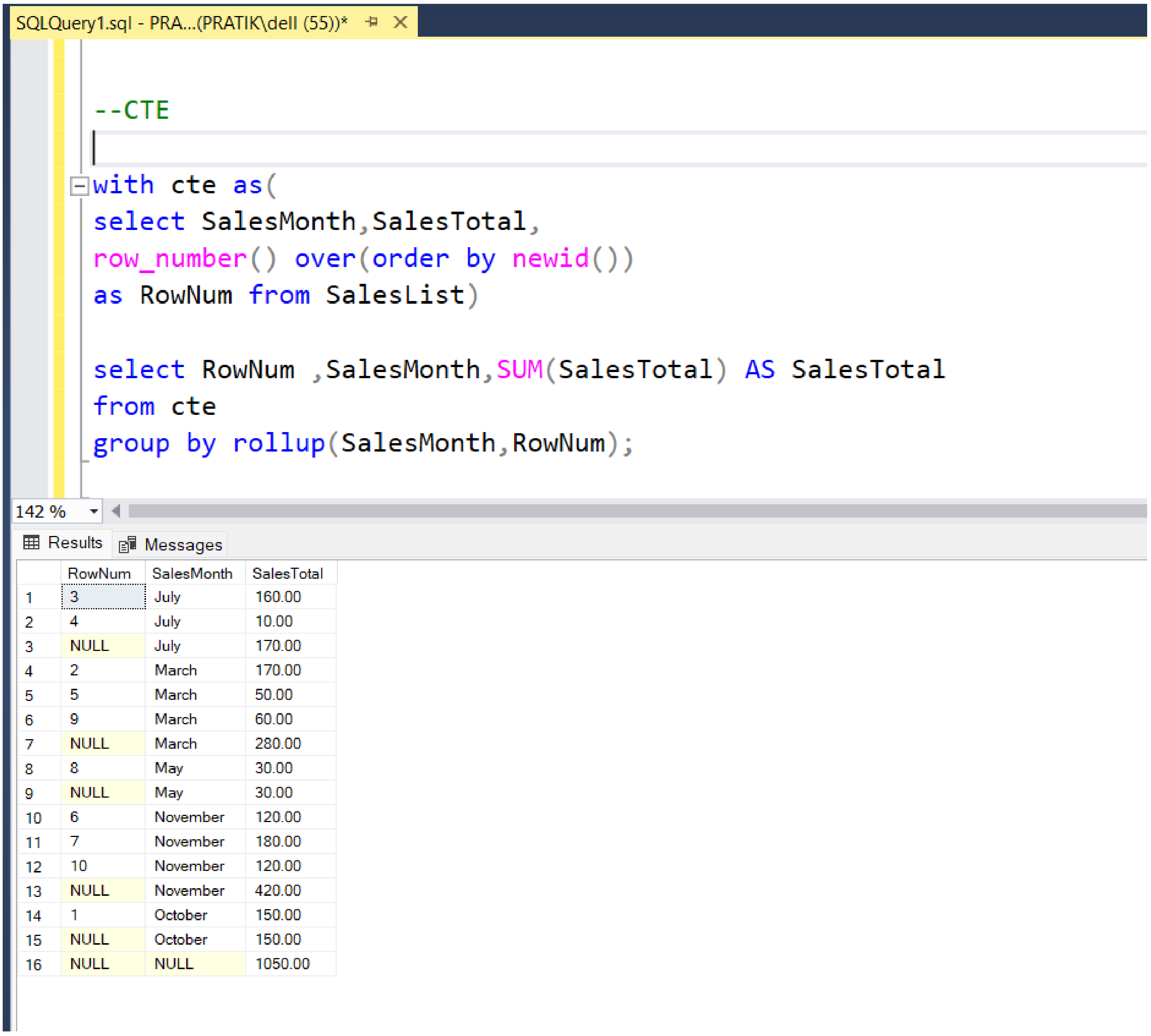
|  | SalesYear | SalesQuartes | SalesMonth | SalesTotal |
|---|---|---|---|---|
| 1 | 2019 | Q1 | March | 60.00 |
| 2 | 2019 | Q1 | NULL | 60.00 |
| 3 | 2019 | Q2 | May | 30.00 |
| 4 | 2019 | Q2 | NULL | 30.00 |
| 5 | 2019 | Q3 | July | 160.00 |
| 6 | 2019 | Q3 | NULL | 160.00 |
| 7 | 2019 | Q4 | November | 300.00 |
| 8 | 2019 | Q4 | October | 150.00 |
| 9 | 2019 | Q4 | NULL | 450.00 |
| 10 | 2019 | NULL | NULL | 700.00 |
| 11 | 2020 | Q1 | March | 220.00 |
| 12 | 2020 | Q1 | NULL | 220.00 |
| 13 | 2020 | Q3 | July | 10.00 |
| 14 | 2020 | Q3 | NULL | 10.00 |
| 15 | 2020 | Q4 | November | 120.00 |
| 16 | 2020 | Q4 | NULL | 120.00 |
| 17 | 2020 | NULL | NULL | 350.00 |
| 18 | NULL | NULL | NULL | 1050.00 |



```sql
--Group by Extension
--using rollup

select  SalesYear,SalesQuartes,SalesMonth ,sum(SalesTotal) as SalesTotal
from SalesList group by rollup(SalesYear, SalesQuartes, SalesMonth);

SELECT SalesYear,SalesQuartes,SUM(SalesTotal) AS SalesTotal ,
GROUPING(SalesQuartes) AS SalesQuarterGrp,
GROUPING(SalesYear) AS SYearGrp
FROM SalesList
GROUP BY ROLLUP(SalesYear, SalesQuartes);
```

| | SalesYear | SalesQuartes | SalesTotal | SalesQuarterGrp | SYearGrp |
|---|---|---|---|---|---|
| 1 | 2019 | Q1 | 60.00 | 0 | 0 |
| 2 | 2019 | Q2 | 30.00 | 0 | 0 |
| 3 | 2019 | Q3 | 160.00 | 0 | 0 |
| 4 | 2019 | Q4 | 450.00 | 0 | 0 |
| 5 | 2019 | NULL | 700.00 | 1 | 0 |
| 6 | 2020 | Q1 | 220.00 | 0 | 0 |
| 7 | 2020 | Q3 | 10.00 | 0 | 0 |
| 8 | 2020 | Q4 | 120.00 | 0 | 0 |
| 9 | 2020 | NULL | 350.00 | 1 | 0 |
| 10 | NULL | NULL | 1050.00 | 1 | 1 |

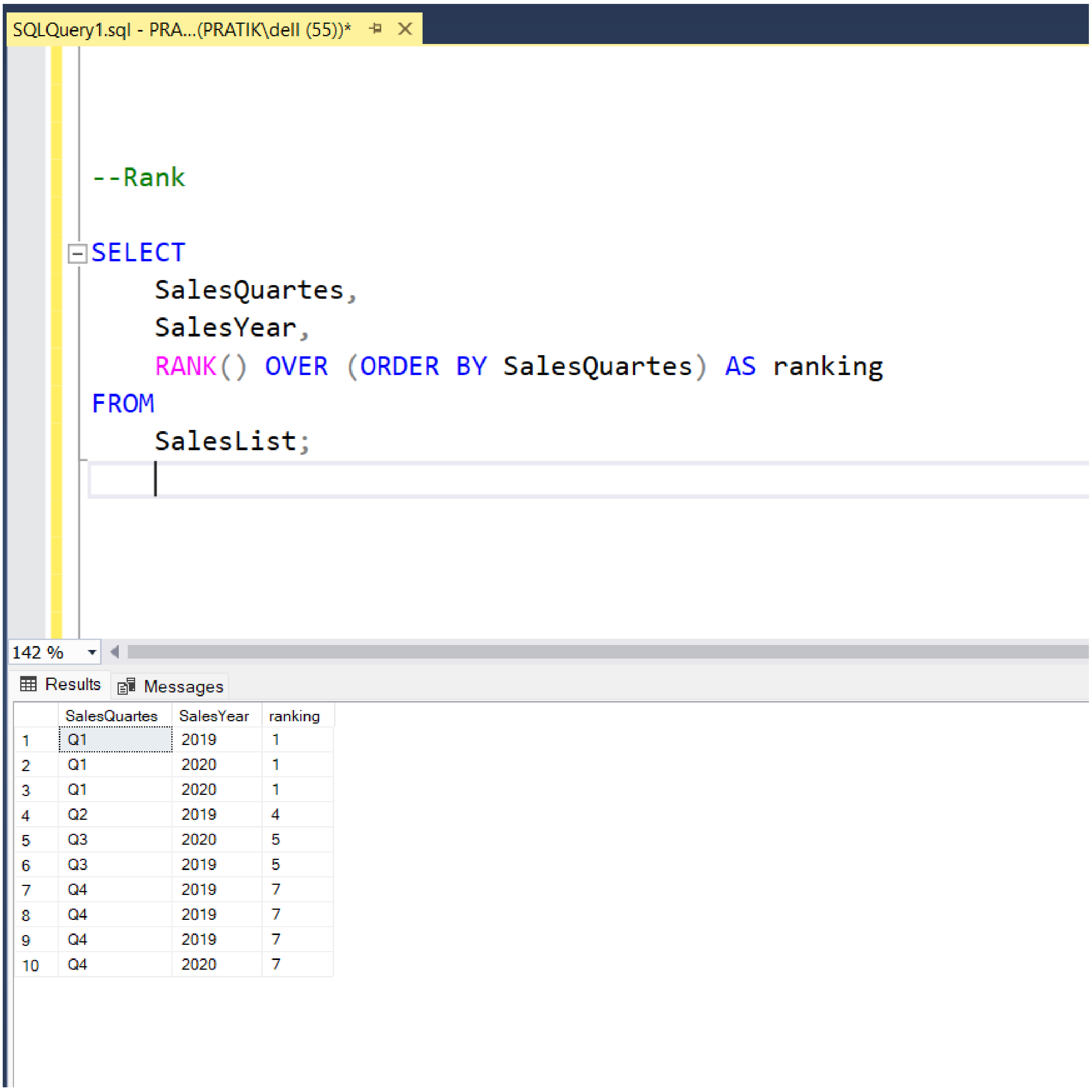
- CTE



```
--CTE

with cte as(
select SalesMonth,SalesTotal,
row_number() over(order by newid())
as RowNum from SalesList)

select RowNum ,SalesMonth,SUM(SalesTotal) AS SalesTotal
from cte
group by rollup(SalesMonth,RowNum);
```

| | RowNum | SalesMonth | SalesTotal |
|---|---|---|---|
| 1 | 3 | July | 160.00 |
| 2 | 4 | July | 10.00 |
| 3 | NULL | July | 170.00 |
| 4 | 2 | March | 170.00 |
| 5 | 5 | March | 50.00 |
| 6 | 9 | March | 60.00 |
| 7 | NULL | March | 280.00 |
| 8 | 8 | May | 30.00 |
| 9 | NULL | May | 30.00 |
| 10 | 6 | November | 120.00 |
| 11 | 7 | November | 180.00 |
| 12 | 10 | November | 120.00 |
| 13 | NULL | November | 420.00 |
| 14 | 1 | October | 150.00 |
| 15 | NULL | October | 150.00 |
| 16 | NULL | NULL | 1050.00 |

- Rank Function:



```
--Rank

SELECT
    SalesQuartes,
    SalesYear,
    RANK() OVER (ORDER BY SalesQuartes) AS ranking
FROM
    SalesList;
```

| | SalesQuartes | SalesYear | ranking |
|---|---|---|---|
| 1 | Q1 | 2019 | 1 |
| 2 | Q1 | 2020 | 1 |
| 3 | Q1 | 2020 | 1 |
| 4 | Q2 | 2019 | 4 |
| 5 | Q3 | 2020 | 5 |
| 6 | Q3 | 2019 | 5 |
| 7 | Q4 | 2019 | 7 |
| 8 | Q4 | 2019 | 7 |
| 9 | Q4 | 2019 | 7 |
| 10 | Q4 | 2020 | 7 |